New algorithms for the analysis of live-cell images acquired in phase contrast microscopy

Thèse

Pierre-Marc Juneau

Doctorat en génie chimique

Philosophiæ doctor (Ph.D.)

Québec, Canada

© Pierre-Marc Juneau, 2015

Résumé

La détection et la caractérisation automatisée des cellules constituent un enjeu important dans de nombreux domaines de recherche tels que la cicatrisation, le développement de l'embryon et des cellules souches, l'immunologie, l'oncologie, l'ingénierie tissulaire et la découverte de nouveaux médicaments. Étudier le comportement cellulaire *in vitro* par imagerie des cellules vivantes et par le criblage à haut débit implique des milliers d'images et de vastes quantités de données. Des outils d'analyse automatisés reposant sur la vision numérique et les méthodes non-intrusives telles que la microscopie à contraste de phase (PCM) sont nécessaires. Comme les images PCM sont difficiles à analyser en raison du halo lumineux entourant les cellules et de la difficulté à distinguer les cellules individuelles, le but de ce projet était de développer des algorithmes de traitement d'image PCM dans Matlab® afin d'en tirer de l'information reliée à la morphologie cellulaire de manière automatisée. Pour développer ces algorithmes, des séries d'images de myoblastes acquises en PCM ont été générées, en faisant croître les cellules dans un milieu avec sérum bovin (SSM) ou dans un milieu sans sérum (SFM) sur plusieurs passages.

La surface recouverte par les cellules a été estimée en utilisant un filtre de plage de valeurs, un seuil et une taille minimale de coupe afin d'examiner la cinétique de croissance cellulaire. Les résultats ont montré que les cellules avaient des taux de croissance similaires pour les deux milieux de culture, mais que celui-ci diminue de façon linéaire avec le nombre de passages. La méthode de transformée par ondelette continue combinée à l'analyse d'image multivariée (UWT-MIA) a été élaborée afin d'estimer la distribution de caractéristiques morphologiques des cellules (axe majeur, axe mineur, orientation et rondeur). Une analyse multivariée réalisée sur l'ensemble de la base de données (environ 1 million d'images PCM) a montré d'une manière quantitative que les myoblastes cultivés dans le milieu SFM étaient plus allongés et plus petits que ceux cultivés dans le milieu SSM. Les algorithmes développés grâce à ce projet pourraient être utilisés sur d'autres phénotypes cellulaires pour des applications de criblage à haut débit et de contrôle de cultures cellulaires.

Abstract

Automated cell detection and characterization is important in many research fields such as wound healing, embryo development, immune system studies, cancer research, parasite spreading, tissue engineering, stem cell research and drug research and testing. Studying *in vitro* cellular behavior via live-cell imaging and high-throughput screening involves thousands of images and vast amounts of data, and automated analysis tools relying on machine vision methods and non-intrusive methods such as phase contrast microscopy (PCM) are a necessity. However, there are still some challenges to overcome, since PCM images are difficult to analyze because of the bright halo surrounding the cells and blurry cell-cell boundaries when they are touching. The goal of this project was to develop image processing algorithms to analyze PCM images in an automated fashion, capable of processing large datasets of images to extract information related to cellular viability and morphology. To develop these algorithms, a large dataset of myoblasts images acquired in live-cell imaging (in PCM) was created, growing the cells in either a serum-supplemented (SSM) or a serum-free (SFM) medium over several passages. As a result, algorithms capable of computing the cell-covered surface and cellular morphological features were programmed in Matlab®.

The cell-covered surface was estimated using a range filter, a threshold and a minimum cut size in order to look at the cellular growth kinetics. Results showed that the cells were growing at similar paces for both media, but their growth rate was decreasing linearly with passage number. The undecimated wavelet transform multivariate image analysis (UWT-MIA) method was developed, and was used to estimate cellular morphological features distributions (major axis, minor axis, orientation and roundness distributions) on a very large PCM image dataset using the Gabor continuous wavelet transform. Multivariate data analysis performed on the whole database (around 1 million PCM images) showed in a quantitative manner that myoblasts grown in SFM were more elongated and smaller than cells grown in SSM. The algorithms developed through this project could be used in the future on other cellular phenotypes for high-throughput screening and cell culture control applications.

Table of Contents

Résumé iii
Abstract v
Table of Contents vii
Tables list xi
Figures list xiii
Abbreviations list xxi
Remerciements xxix
Foreword xxxi
Chapter 1. Introduction 1
Chapter 2. Literature Review 7
2.1 High throughput Screening (HTS) 7
2.2 Culture of myoblasts 11
2.2.1 Cell culture 11
2.2.2 Culture medium 14
2.2.3 Cellular phenotypes associated to the myogenic lineage 15
2.2.4 Cell morphology 23
2.2.5 Cellular growth 30
2.3 Microscopy 34
2.3.1 Types of microscopy 34
2.3.2 Controlled environment for live cell imaging 43
2.3.3 Cell culture sample integrity 44
2.3.4 Microscopic image acquisition 48
2.4 Image processing 56
2.4.1 Definition of a digital image 56
2.4.2 Fourier transform of a digital image 58
2.4.3 Image pre-processing 62
2.4.4 Segmentation algorithms 68
2.4.5 Pattern recognition 83
2.4.6 Wavelet transforms 95
2.4.7 Classifiers 112
2.4.8 Parameters calibration 113
2.4.9 Commercial products/softwares for image processing 121

2.5 Multivariate data analysis 125
2.5.1 Matrix of features 125
2.5.2 Multivariate techniques 126
Chapter 3. Materials and Methods 141
3.1 Cell culture 141
3.2 Microscopy 141
3.3 Image acquisition 145
3.4 Image processing and data analysis 146
Chapter 4. Selection and tuning of a fast and simple phase contrast microscopy image segmentation algorithm for measuring myoblasts growth kinetics in an automated fashion 147
4.1 Résumé 147
4.2 Abstract 148
4.3 Introduction 149
4.4 Materials and methods 151
4.4.1 Proposed segmentation algorithm 151
4.4.2 Tuning of segmentation algorithms 155
4.4.3 Kinetic growth models 156
4.4.4 Experimental set-up 156
4.5 Results 157
4.5.1 Selecting the objective function 157
4.5.2 Comparative performance of the segmentation algorithms 159
4.6 Discussion 163
4.6.1 Application 1: cell-covered surface (S) growth curves 163
4.6.2 Application 2: Correlation between kinetic parameters and culture conditions (passage number) 166
4.7 Conclusion 168
4.8 Ackowledgements 169
Chapter 5. The Undecimated Wavelet Transform - Multivariate Image Analysis (UWT-MIA) for simultaneous extraction of spectral and spatial information 171
5.1 Résumé 171
5.2 Abstract 172
5.3 Introduction 173
5.4 Materials and methods 176

5.5 The Undecimated Wavelet Transform – Multivariate Image Analysis (UWT-MIA) 178
5.5.1 Gabor wavelet planes generation 178
5.5.2 UWT-MIA 182
5.6 Results and Discussion 184
5.6.1 Textural and spectral segmentation of a RGB image (Set 1) 185
5.6.2 Textural and spectral segmentation in multiple RGB images (Set 2) 190
5.6.3 Texture segmentation in multiple cellular images (Set 3) 193
5.7 Conclusions 198
5.8 Acknowledgements 200
Chapter 6. Using the undecimated wavelet transform multivariate image analysis (UWT-MIA) as an on-line tool to characterize adherent cells morphology 201
6.1 Résumé 201
6.2 Abstract 202
6.3 Introduction 203
6.4 Materials and methods 205
6.5 Image processing and data analysis 206
6.5.1 UWT-MIA performed on the calibration dataset 208
6.5.2 Extraction of relevant image features for cellular morphological characterization 213
6.5.3 Estimation of cellular morphological features for the entire PCM image database 217
6.6 Discussion 224
6.7 Conclusion 226
6.8 Acknowledgements 227
Chapter 7. Conclusion and recommendations 229
7.1 Research problem 229
7.2 Contributions 231
7.3 Perspectives 234
References 237
Appendix A. Hyperspectral microscopy 251
A.1 Introduction 251
A.2 Methodology 251
A.2.1 Hyperspectral cameras specifications 251

A.2.2 Control of automated stage and hyperspectral cameras 253
A.2.3 Estimation of spatial resolutions Δx and Δy 254
A.2.4 Light sources 255
A.3 Experiments description 257
A.4 Phase contrast hyperspectral microscopy with VIS camera (PCM-VIS) 258
A.5 Epifluorescence hyperspectral microscopy with VIS camera (EM-VIS) 263
A.6 Phase contrast hyperspectral microscopy with NIR camera (PCM-NIR) 268
A.7 Conclusion 277
Appendix B. Commercial products/software for microscopy, image acquisition and image processing 279
Appendix C. Protocols used for the culture of human myoblasts 289
C.1 Frozen cells thawing and seeding 289
C.2 Cell culture trypsinization and passaging 290
C.3 Cells freezing protocol 291
C.4 Hemacytometer cell count 292
C.5 Preparation of serum-supplemented medium (SSM) 293
C.6 Preparation of serum-free medium (SFM) 294
C.7 Compounds for cell culture 296
C.8 Composition of MCDB120 297
Appendix D. List of PCM image datasets generated during the project 299
Appendix E. Metamorph® image acquisition procedure (IX81 Olympus Microscope) .. 305
E.1 Set cell culture environment 306
E.2 Set the zero position (x=0, y=0) on the automated stage 306
E.3 Generate lists of points (if required) 306
E.4 Program a Metamorph acquisition journal (if required) 307
E.5 Place the cell culture multiwell plate inside the microscope 308
E.6 Adjust manually the focus for each well 309
E.7 Enter the proper values in the "*Move To An Absolute Position"* functions 310
E.8 Set all the settings in the MDA window 311
E.9 Launch the image acquisition 312
E.10 Matlab® script to generate lists of points for MDA (Metamorph®) 313
Appendix F. Matlab® scripts used for image and data processing 317
Appendix G. Matlab® scripts used for hyperspectral microscopy 383

Tables list

Table 2-1 : Labeling methods in HTS (Rimon and Schuldiner 2011) 10
Table 2-2 : List of molecular markers (genetic factors) associated to different stages of skeletal muscle formation (Buckingham, Bajard et al. 2003) 19
Table 2-3 : Molecular markers for satellite cells (Charge and Rudnicki 2004) 20
Table 2-4 : Challenges related with the size analysis of myotubes in microscopic images (Agley, Velloso et al. 2012) 29
Table 2-5 : Short description of the different types of microscopes 36
Table 2-6 : Gaussian wavelet cutoff frequencies and periods (for σ = 0.5) 110
Table 3-1 : Experimental resolutions for each microscope objective 144
Table 4-1: Segmentation parameters obtained by optimization 159
Table 4-2: Algorithms parameters and grid search settings 161
Table 4-3: Performance comparison between algorithms on ground truth images 162
Table 4-4: Linear relationship parameters between kinetic parameters and passage number (for $N_{Passage}$ 5 to 18) 168
Table 5-1 : Gabor wavelet parameters used for each image set 185
Table 6-1 : Optimal mask parameters and PLS models performance for each cellular morphological features distribution estimated using UWT-MIA features. 215
Table A-1 : Hyperspectral cameras specifications 252
Table A-2 : Estimated Δx and Δy for VIS hyperspectral microscopy 255
Table A-3 : Series of experiments realized in hyperspectral microscopy 257
Table C-1 : Serum-supplemented medium (SSM) ingredients 293
Table C-2 : Serum-free medium (SFM) ingredients 294
Table C-3 : List of compounds used for cell culture and their suppliers 296
Table C-4 : Composition of MCDB120 medium (Ham and Blau 1988) 297
Table E-1 : Functions needed for each well *i* 308

Figures list

Figure 2-1 : Whole process behind high throughput screening (HTS) experiments (adapted from Rimon and Schuldiner (2011)). 8
Figure 2-2 : Influence of labeling methods and experimental strategies on the type and quality of information that can be obtained on whole organism or single cell structure and behavior (adapted from Rimon and Schuldiner (2011)). 9
Figure 2-3 : Typical images of myoblasts at low (A) and high (B) confluence, as well as undergoing fusion (C) at a magnification of 10X 13
Figure 2-4 : Illustration of how cells from a biopsy can be cultivated over several passages 13
Figure 2-5 : Typical skeletal muscle with myofibers surrounded by satellite cells (adapted from Charge et al. (2004) and Danoviz et al. (2012)) 16
Figure 2-6 : Different phenotypes in the myogenic lineage as well as relevant molecular markers (adapted from Bentzinger et al. (2012)) 17
Figure 2-7 : Molecular markers associated to the differentiation of stem cells to muscle cells (adapted from Deasy et al. (2001)) 21
Figure 2-8 : Rough description of the cellular morphology through the myogenic lineage (adapted from Rudnicki et al. (2008)) 23
Figure 2-9 : Qualitative characterization of the morphology of adherent cells in PCM (adapted from Theriault et al. (2012)) 24
Figure 2-10 : Shape machine model for cellular morphology (adapted from Satulovsky et al. (2008)) 26
Figure 2-11 : A second lens to redirect the light towards the eye of the observer creates a virtual image much bigger than the original object (adapted from Cox (2007)) 35
Figure 2-12 : Optical path associated to phase contrast microscopy (adapted from Cox (2007)) 39
Figure 2-13 : Effect of specimen thickness on phase difference (adapted from Zeiss (2013)) 41
Figure 2-14 : Acquisition grid and overlap concept 50
Figure 2-15 : Mosaic of 1521 images acquired to image an entire well of a 24 wells multiwell plate in PCM (20X): there is a loss of phase contrast at the well periphery. 52
Figure 2-16 : Examples of images taken at the wrong and right focus (myoblasts imaged at 40X in PCM) 54
Figure 2-17 : Example of an optimal focus profile at the bottom of a well in a 24 wells multiwell plate. At each well location (for x and y coordinates), the Metamorph® autofocus utility found the optimal focus. The optimal focuses are provided as a topographic map, with isolines providing the z-coordinate level (focus plane height). 54
Figure 2-18 : Example of a 2D digital image (myoblasts imaged in PCM at 20X) 56
Figure 2-19 : Example of a hyperspectral image, with two spatial dimensions and one spectral dimension. The information is summarized via a T_1 score image generated by Principal Component Analysis (PCA). 57

Figure 2-20 : Example of a Fourier transform of a 2D digital image of dimension 59
Figure 2-21 : Magnitude, power spectrum and phase angle matrix associated to the DFT of a typical PCM image 60
Figure 2-22 : Phase angle and power spectra matrices for images of cells grown in SSM 61
Figure 2-23 : Phase angle and power spectra matrices for images of cells grown in SFM 62
Figure 2-24 : Spatial filtering using the average filter of size N1 by N2..... 65
Figure 2-25 : Convolution of an image with a linear (Gaussian) filter in the frequency domain 66
Figure 2-26 : Image segmentation process, in which a Segmentation Map (SM) is generated from a raw image 69
Figure 2-27 : Example of the evolution of a level set representation for segmenting a cell. A significant number of iterations is required to achieve a segmentation. 72
Figure 2-28 : Watershed algorithm approach: as water fills catchment basins, pixels in an image are grouped together, starting from minima, until two groups of pixels reach each other, thus detecting a watershed line (dam) (Vincent and Soille 1991). 78
Figure 2-29 : A-Raw image; B-Watershed transform applied directly on a raw image. There is significant over-segmentation, with a large number of tiny objects segmented (one color for each group of pixel segmented). 81
Figure 2-30 : A-Raw image; B-Cellular regions highlighted (green) for object markers; C-Background and object markers identified (red); D- Segmentation results using the watershed transform (one color for each cell). 82
Figure 2-31 : Feature generation process (adapted from Rodenacker and Bengtsson (2003))..... 84
Figure 2-32 : Different classes of features (adapted from Rodenacker and Bengtsson (2003))..... 85
Figure 2-33 : GLCM technique 88
Figure 2-34 : Frequency content (scale) and orientation content information in the FFT of an image 91
Figure 2-35 : Shape-based features generated from a segmentation map..... 93
Figure 2-36 : Round cells localized by template convolution 94
Figure 2-37 : FWT approach to compute the DWT for signal decomposition and synthesis (adapted from Gonzalez and Woods (2008)) 97
Figure 2-38 : Sinusoidal 1D signal (A) submitted to a DWT (B) or a CWT(C) decomposition. Detail coefficients for both decompositions are shown using a color map: strong positive values are in white and strong negative values are in black. 98
Figure 2-39 : FWT approach to compute the DWT for 2D signal decomposition and synthesis (adapted from Gonzalez and Woods (2008)) 101
Figure 2-40 : 2D-DWT applied to a typical cell image 102
Figure 2-41 : Impact of σ, ε and k_0 on the shape of a typical Gabor wavelet 104
Figure 2-42 : UWT-CWT decomposition using the Gaussian wavelet on a typical cell image 106

Figure 2-43 : UWT-CWT decomposition using the Gabor wavelet on a typical cell image 107
Figure 2-44 : Relationship between frequency resolution, space resolution and wavelet scale..... 111
Figure 2-45 : Features space seen as an *n*-dimensional Euclidean space..... 113
Figure 2-46 : Typical confusion matrix for a segmentation algorithm used to differentiate cellular regions from background regions 116
Figure 2-47 : Example of ROC curves for comparing two algorithms for classification or segmentation 119
Figure 2-48 : Live-cell imaging/HTS steps for which several products/softwares offer a solution..... 123
Figure 2-49 : One component PCA model capturing the variation in a 3 dimensions features space in which a data point x_i is projected onto p_1 ($t_{i,1}$) (adapted from Wold, Esbensen et al. (1987)) 127
Figure 2-50 : Size of the different matrices involved in PCA (adapted from Wold, Esbensen et al. (1987))..... 128
Figure 2-51 : Size of the different matrices involved in PLS (adapted from Wold, Geladi et al. (1987) and from Wold, Sjostrom et al. (2001))..... 131
Figure 2-52 : Houses RGB image (Kingdom and Olmos) and corresponding RGB cube. It is possible to analyze the RGB cube slice by slice (X_i)..... 134
Figure 2-53 : p_1, p_2 and p_3 loadings from MIA performed on the Houses RGB image (Kingdom and Olmos)..... 135
Figure 2-54 : T_1, T_2 and T_3 score images from MIA performed on the Houses RGB image (Kingdom and Olmos)..... 135
Figure 2-55 : T_1/T_2, T_1/T_3 and T_2/T_3 score density histograms from MIA performed on the Houses RGB image (Kingdom and Olmos). 136
Figure 2-56 : T_1/T_2 masking results (selected regions in cyan) from MIA performed on the Houses RGB image (Kingdom and Olmos). 137
Figure 2-57 : PRESS as a function of the number of components *A* (adapted from Geladi and Kowalski (1986)) 140
Figure 3-1 : Qimaging® monochrome camera attached to the side port of the IX81 Olympus™ microscope 142
Figure 3-2 : IX81 Olympus™ microscope with enclosure 143
Figure 3-3 : IX81 Olympus™ microscope resolution in PCM from hemacytometer images. 143
Figure 3-4 : CO_2 and temperature control of the experimental set-up 144
Figure 3-5 : CO_2 enclosure with ports for CO_2/air mix (secondary enclosure). This enclosure can be fitted on the automated stage. 145
Figure 4-1: Generation of the Range Map (RM) with a range filter window (RFW) of size RFS (range filter size) 152
Figure 4-2: The segmentation method illustrated for a specific line of pixels in a phase-contrast microscopy image. A: Raw image and a selected line of pixels (black). B: Raw pixel values of a selected line. C: Range Map (RM) generated by a

range filter. D: Cellular region binary mask (CRBM) calculated from RM. E: Image with cellular regions (white) highlighted. 153
Figure 4-3: Examples of ground truth images used. A: Raw images. B: Result of the optimized segmentation algorithm. White is true positive, grey is false positive/false negative, and black is true negative. 157
Figure 4-4: Example of the segmentation algorithm performance for different optimization criteria: (A) original image; (B)-(E) maximization of the F function for (B): α=0, (C): α=0.5, (D): α=0.7 and (E): α=1; (F) minimizing the area estimated error (AEE). White pixels represent true positive, grey pixels represent false positive/false negative, and black pixels represent true negative. 158
Figure 4-5: A: Typical images of the evolution of a culture (from low to high cell-covered area); B: Evolution of S against time for each position in the acquisition grid, and average cell-covered surface (white dots) as a function of time; C: Standard deviation of S against culture time; Distribution of the individual cell-covered surfaces at (D): 0h, (E): 58h, and (F): 162h. 163
Figure 4-6: Fitting of the kinetic models on the average cell-covered surface. Effect of passage number on growth kinetics of cells grown in serum-supplemented medium (SSM) for different kinetic models (A) Exponential; (B) Logistic. Comparison between SSM and serum-free medium (SFM) at $N_{passage}$=6 for different kinetic models (C) Exponential; (D) Logistic. Effect of FBS quality on cells growth kinetics for different kinetic models (E) Exponential; (F) Logistic. 165
Figure 4-7: Kinetic parameters for each model as a function of $N_{Passage}$ meeting all criteria. (A) μ for the exponential model; (B) C_{max} for the logistic model. 167
Figure 5-1 : Images used to illustrate the UWT-MIA approach. A) Set 1: one synthetic textured RGB image; B) Set 2: six synthetic RGB images with colored circular objects; C) Set 3: six phase contrast microscopy images of myoblasts. 176
Figure 5-2 : Steps associated with the UWT-MIA method 178
Figure 5-3 : The 1D and 2D shape of the Gabor wavelet for a given set of parameters and orientation angles (θ = 0, π/4 and 2π/4). White pixels are associated with positive filter weights and black pixels to negative filter weights. 181
Figure 5-4 : Generation and stacking of the wavelet planes (WP) obtained for each wavelength (λ), angle (θ), and scale (s) into a cube (WPC)..... 183
Figure 5-5 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 1 image 186
Figure 5-6 : Investigation of the pixel clusters in T_1/T_2 score density histogram. A) Raw image showing colored stripes and edges. Localization of pixels belonging to B) colored stripes and edge pixels and C) vertical, horizontal and diagonal patterns. D) Clustering of blue/green and yellow pixels. 187
Figure 5-7 : Pseudo-color images of WPs for Set 1 for λ = Red, s = 10 and for A) θ = 0, B) θ = π/4, C) θ = 2π/4, D) θ = 3π/4. In the color map, red/yellow colors correspond to positive wavelet coefficients and blue/green colors to negative wavelet coefficients. 188
Figure 5-8 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in red) for Set 1. Masks are used to select

A) diagonal, B) horizontal, C) vertical, D) yellow, and E) blue/green pixel regions. 189
Figure 5-9 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 2 images 191
Figure 5-10 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in cyan) for Set 2. Selected pixels correspond to A) background region, B) red objects, C) blue objects, D) small red objects, and E) large red objects. 192
Figure 5-11 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 3 images 194
Figure 5-12 : Investigation of the pixel clusters in T_1/T_2 score density histogram. A) raw image showing the cells cytoplasm and contour pixels. Localization of pixels belonging to B) cell contour and cytoplasm, C) vertical, horizontal, diagonal and background pixels. 194
Figure 5-13 : T_1/T_2 individual score density histograms for three images of Set 3. A) Raw images, B) corresponding individual score density histograms. The effect of the degree of confluence (number of cells) and the cells orientation on the T_1/T_2 score density histogram are shown. 196
Figure 5-14 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in red) for Set 3. Selected pixels correspond to A) background regions, B) vertical cells, C) horizontal cells, D) cellular contours, and E) cells cytoplasm. 197
Figure 6-1 : Major steps involved in the proposed pattern recognition approach to estimate and analyze cellular morphological features. Using wavelet planes cubes (WPCs), it is possible to generate score density histograms (SDHs) by performing a principal component analysis (PCA) and to compute morphological features via Partial Least Squares (PLS) models. 207
Figure 6-2 : Loadings for PCA performed on the WPCs of the 200 calibration images. The blue and red lines correspond to the first and second components, respectively. 209
Figure 6-3 : A) T_1/T_2 score density histogram of the global PCA model built from the 200 calibration images. The color scale (from black to white) indicates the density of pixels having a certain combination of T_1/T_2 values. Pixels associated with B) background regions C) cell of different sizes, and D) orientations. 209
Figure 6-4 : Images and their SDH for low (A) or high (B) cell confluence, for round cells (C) and for elongated cells oriented vertically (D) or horizontally (E). 211
Figure 6-5 : Illustration of the masking procedure in the T_1/T_2 score density histograms. Three different masks are applied (cyan color) on pixel regions associated with the background region (A), round or large cells (B) and small or elongated cells (C). The identified pixels are then overlaid onto three PCM images. 212
Figure 6-6 : Parameters associated with the series of masks used for estimating A) cell size related distributions and B) cell orientation distribution. 214
Figure 6-7 : Performance of PLS models for estimating cellular morphological features distributions illustrated on two calibration images. Black bars are manual cell counts (ground truth data) and hatched bars are cell counts estimated by the PLS models. 216

Figure 6-8 : Average CCSR and average cell count curves (sum of individual minor axis bins) for a typical SSM and SFM well. 218
Figure 6-9 : Average major axis (A), minor axis (B), orientation (C) and roundness (D) curves for a typical SSM and SFM well. SSM curves are in red and SFM curves are in blue. 219
Figure 6-10 : Average CCSR and average cell count curves (sum of individual minor axis bins) (A), major axis (B) and minor axis (C) for two SSM wells with FBS1 (red) and FBS2 (blue). 220
Figure 6-11 : : T_1/T_2 scores scatter plots of the PCA model built on the morphological features extracted on the entire PCM image dataset. A) Points are colored according to CCSR, from low (blue) to high (red) values. B) Points colored according to the culture medium used, red (SSM) and blue (SFM). 221
Figure 6-12 : Loadings of the first (black) and second (red) PCA components. Maj corresponds to the major axes (in pixels), Min corresponds to the minor axes (in pixels) and Round corresponds to roundness. 222
Figure 6-13 : Typical images of myoblasts grown in SSM and SFM at low and high CCSR (at confluence). 223
Figure A-1 : UV, VIS and NIR hyperspectral cameras available on the bench (can be fitted on the IX81 microscope lateral port) 251
Figure A-2 : VIS and NIR camera calibration 253
Figure A-3 : Brightfield hyperspectral T_1 score image of a hemacytometer at 10X, with Δx = 1 µm between slices 254
Figure A-4 : Transmittance as a function of light wavelength (in nm) for the Microscope light source tested by the COPL. Wavelengths higher than 750 nm are blocked. 256
Figure A-5 : λ and y slices images acquired through the IX81 Microscope with A-Microscope light source (imaging a hemacytometer); B-Bench light source. The bench light source does not block wavelengths at 750 nm. 256
Figure A-6 : p_1 and p_2 loadings from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013 259
Figure A-7 : T_1/T_2 global score density histogram from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013 259
Figure A-8 : T_1 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013 260
Figure A-9 : T_2 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013 261
Figure A-10 : T_1/T_2 individual score density histograms from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013 262
Figure A-11 : T_1 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment November 8th 2012 262
Figure A-12 : p_1 and p_2 loadings from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013 263
Figure A-13 : T_1/T_2 global score density histogram from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013 264

Figure A-14 : T_1 score images from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013 265
Figure A-15 : T_2 score images from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013 266
Figure A-16 : T_1/T_2 individual score density histograms from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013 267
Figure A-17 : p_1, p_2 and p_3 loadings from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 269
Figure A-18 : T_1/T_2 global score density histogram from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 269
Figure A-19 :T_2/T_3 global score density histogram from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 270
Figure A-20 : T_1 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 271
Figure A-21 : T_2 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 272
Figure A-22 : T_3 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 273
Figure A-23 : T_1/T_2 individual score density histograms from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 274
Figure A-24 : T_2/T_3 individual score density histograms from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013 275
Figure A-25 : T_2/T_3 masking and overlays to segment background pixels (PCM-NIR hyperspectral cubes for experiment January 6th 2013) 276
Figure A-26 : T_2/T_3 masking and overlays to segment cellular pixels (PCM-NIR hyperspectral cubes for experiment January 6th 2013) 276
Figure E-1 : Metamorph® interface 305
Figure E-2 : Apps menu to open MDA window 306
Figure E-3 : How to program a Metamorph acquisition journal 307
Figure E-4 : RESET FOCUS journal 309
Figure E-5 : Device Control and MDA windows (live acquisition) to adjust manually the focus for each well 309
Figure E-6 : Example of focuses differences computed between adjacent wells for a 6 wells multiwell plate 310
Figure E-7 : The *Move to an Absolute Position* value (starting from well #2) needs to be modified to incorporate the focuses differences obtained in Section E.6. 310
Figure E-8 : MDA window with different options/settings 311
Figure E-9 : MDA journal for image acquisition at fixed intervals 312
Figure E-10 : Examples of images acquired (typical name structure) 313

Abbreviations list

ACC	Classification Accuracy
ACT	Average Computing Time per image
AEE	Error on Estimated Area for cells
AIC	Akaike Information Criterion
AMT	Angle Measure Technique
AUC	Area Under the Curve
BF	Brightfield Microscopy
BPAE	Bovine Pulmonary Artery Endothelial cells
CCD	Charge-Coupled Device
CCSR	Cell-Covered Surface Ratio
CHO	Chinese Hamster Ovary cells
CM	Confocal Microscopy
CRBM	Cellular Region Binary Mask
CRF	Conditional Random Fields
CV	Cross-Validation
CWT	Continuous Wavelet Transform
DF	Darkfield Microscopy
DFM	Differenciation Medium
DFT	Discrete Fourier Transform
DHM	Digital Holography Microscopy
DIC	Differential Interference Contrast Microscopy
DOE	Design of Experiment
DoG	Difference-of-Gaussians
DT	Doubling Time
DWT	Discrete Wavelet Transform

EFM	Epifluorescence Microscopy
EM	Epifluorescence Microscopy
FA	Factor Analysis
FD	Fractal Dimension
FFT	Fast Fourier Transform
FLD	Fisher Linear Discriminant
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FWT	Fast Wavelet Transform
GLCM	Grey Level Co-occurrence Matrix
HCRF	Hidden Conditional Random Fields
HCS	High Content Screening
HL	Holography
HTS	High-Throughput Screening
IA	Image Acquisition
ICA	Independent Component Analysis
ICM	Iterated Conditional Modes
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IMS	Image Mapping Spectrometer
ISM	*In-situ* Microscope
ISSO	Immersion Simulation based Self-Organizing transform
KF	Kalman Filter
KNN	K-Nearest Neighbor
KS	Kolmogorov-Smirnov test

LDA	Linear Discriminant Analysis
LLE	Locally-Linear Embedding
MCR	Multivariate Curve Resolution
MDA	Multiple Dimension Acquisition
MIA	Multivariate Image Analysis
MIR	Multivariate Image Regression
MIX	Multivariate Image Texture Analysis
MLR	Multiple Linear Regression
MOST	Minimum Object Size Threshold
MPC	Muscle Precursor Cell
MPM	Multiphotons Microscopy
MPS	Multipurpose Software
MRF	Markov Random Field
MR-MIA	Multi-Resolution Multivariate Image Analysis
MRT	Minimum Range Threshold
NIPALS	Nonlinear Iterative Partial Least Squares
NN	Neural Networks
NPV	Negative Predictive Value
PC	Principal Component
PCA	Principal Component Analysis
PCM	Phase Contrast Microscopy
PLS	Partial-Least Squares
PLS-DA	Partial-Least Squares Discriminant Analysis
PMT	Photomultiplier
PPV	Positive Predictive Value
PR	Precision-Recall (curve)

PRESS	Prediction Sum of Squares
QD	Quantum Dot
RFS	Range Filter Size
RFW	Range Filter Window
RM	Range Map
RMSECV	Root Mean Square Error of Cross-Validation
ROC	Receiver Operator Characteristic
SDH	Score Density Histogram
Se	Sensitivity
SE	Structuring Element
SFFS	Sequential Forward Floating Selection
SFM	Serum-Free Medium
SIFT	Scale-Invariant Feature Transform
SLS	Separable Least Squares
SM	Segmentation Map
SNR	Signal-to-Noise Ratio
SOM	Self-Organizing Map
Sp	Specificity
SSE	Sum of Squares of Errors
SSM	Serum-Supplemented Medium
SSR	Sum of Squares of Regression
SST	Sum of Squares Total
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SWT	Stationary Wavelet Transform
TN	True Negative

TP	True Positive
TPR	True Positive Rate
UNN	Unique Nearest Neighbor
US	Unspecified
UWT	Undecimated Wavelet Transform
UWT-MIA	Undecimated Wavelet Transform Multivariate Image Analysis
WFLT-LCI	Wide-Field Long-Term Live Cell Imaging
WP	Wavelet Plane
WPC	Wavelet Planes Cube
WTA	Wavelet Texture Analysis

À ma grand-mère Lucille

Même dans la maladie ton courage et ta persévérance nous ont tous inspirés

Merci pour tout

Remerciements

J'aimerais remercier chaleureusement mon directeur de recherche (Dr. Carl Duchesne) et mon co-directeur de recherche (Dr. Alain Garnier) pour leur aide, leur soutien et leur présence. Comme dans tout projet d'études graduées il y a toujours des hauts et des bas (analogie ici au domaine fréquentiel ☺), mais vous m'avez toujours encouragé à poursuivre et à me dépasser. Je vous remercie d'avoir cru en moi et de m'avoir donné la chance de faire des études graduées dans un domaine difficile mais intéressant au point de vue scientifique.

Un grand merci au Dr. Jacques P. Tremblay, à M. Victor Parent et à Mme Marlyne Goulet pour leur aide et leur expertise dans la culture de myoblastes humains. Cultiver des cellules humaines *in vitro* est en soi une discipline complexe (qui m'était tout à fait inconnue), et sans votre aide ce projet n'aurait pas eu lieu. Mes sincères remerciements au Prof. J. Jay Liu pour nous avoir aidé avec les ondelettes Gausiennes continues, et à l'équipe de Calcul-Québec (Colosse) pour nous avoir permis de calculer les résultats de la base de données d'images dans des temps acceptables.

Merci à ma conjointe (mon amour) Catherine qui m'a toujours encouragé et aidé dans cette aventure. Sans ton aide et ton soutien ce projet de doctorat n'aurait pas été possible! Merci à ma famille de leur soutien : mes parents Pierre et Carole, mon grand-père Roger, Lucille, Jean, Charles-Philipp, Ann-Catherine et à toute la famille Juneau, Frenette, Taillon et de Varennes. Merci aussi à Suzanne et à la famille Dumas de leur soutien. J'aimerais souligner le support de la compagnie Soutex (Michel, Robert, Sylvain, Daniel et tous les collègues) au cours de mon doctorat. J'ai beaucoup appris en tant qu'ingénieur et en tant que personne à votre contact. Un gros merci!

Mes remerciements aux étudiants, stagiaires et postdocs pour leur support, leurs encouragements et les discussions intéressantes. Au risque d'oublier certains et certaines qui sont passés au cours des années, merci à Julien, Massoud, Francis, Olivia, Wilinthon, Corinne, Igor, Mathias, Jean-Pascal, Imad, Hanan, Marième, Myriam, Ryan, Vanessa, Simon, Thiery, Patrick, Sarah, Moez, Jaya, Amélie, Karl, Juliette, Alexandre, Jean-Philippe, Zahraa, Younes. Mes plus sincères remerciements aux membres du personnel du département de génie chimique qui sont toujours là pour nous épauler dans nos projets: Jean-Nicolas, Jérôme, Marc, Lise, Yann, Pierrette, Nadia et Ann.

Foreword

Pierre-Marc Juneau (PhD candidate) performed the laboratory work (myoblasts cultivation and microscopic image acquisition) as well as computer programming, image processing and data analysis. Pierre-Marc Juneau led the writing of the three papers presenting the main results and contributions of this PhD project. The three papers are presented in Chapter 4, Chapter 5 and Chapter 6. Prof. Carl Duchesne and Prof. Alain Garnier contributed as coauthors to the design of experiments and to the revision of the papers. Two papers are already published or accepted and one will be submitted in 2015:

- **Juneau, P. M., A. Garnier, and C. Duchesne (2013). "Selection and Tuning of a Fast and Simple Phase-Contrast Microscopy Image Segmentation Algorithm for Measuring Myoblast Growth Kinetics in an Automated Manner." Microscopy and Microanalysis 19(4): 855-866.**

- **Juneau, P.-M., A. Garnier, and C. Duchesne (2015). "The undecimated wavelet transform–multivariate image analysis (UWT-MIA) for simultaneous extraction of spectral and spatial information." Chemometrics and Intelligent Laboratory Systems 142(0): 304-318.**

- **Juneau, P.-M., A. Garnier, and C. Duchesne. " Using the undecimated wavelet transform multivariate image analysis (UWT-MIA) as an on-line tool to characterize adherent cells morphology", to be submitted to Biotechnology and Bioengineering Journal in 2015.**

As part of another project led by a post-doctorate student (Dr. Corinne Hoesli), several Matlab® codes developed for the three papers were reused to analyze live-cell images of endothelial progenitor cells. The results are provided in a paper (not included in this thesis) :

- **Hoesli, C. A., Garnier, A., Juneau, P. M., Chevallier, P., Duchesne, C. and Laroche, G. (2014). "A fluorophore-tagged RGD peptide to control endothelial cell adhesion to micropatterned surfaces." Biomaterials 35: 879-890.**

Several aspects and results related to the image processing codes developed for this PhD thesis were presented in seminars and conferences:

- **Juneau P.M.**, Garnier A., Duchesne C., *Online monitoring of myoblast cell cultures using hyperspectral imaging*, Quebec-Ontario Biotech meeting, June 3-4 2010, Montréal (Polytechnique).
- **Juneau P.M.**, Duchesne C., Garnier A., *Cell Culture Monitoring Using Hypespectral Microscopy*, 60th Canadian Chemical Engineering Conference, Saskatoon, Canada, 2010.
- Juneau P.M., Duchesne C., Garnier A., *Using phase contrast microscopy to monitor myoblast cell cultures*, 10ième Colloque de l'Association de Thérapie Génique du Québec (ATGQ), Université Laval, Québec, Canada, November 18 2011 (poster presentation).
- Juneau P.M., Duchesne C., Garnier A., *Phase contrast microscopy image segmentation and analysis*, Cell Culture Engineering XIII (CCEXIII), April 22-27 2012, Scottsdale, Arizona, USA (poster presentation).
- Juneau P.M., Garnier A., Duchesne C., *Automated monitoring of myoblast cultures using phase contrast microscopy*, 1st Till & McCulloch meetings, April 30 – May 2 2012, Montreal, Canada (poster presentation).
- **Juneau P.M.**, Garnier A., Duchesne C., *Estimation of the state (health) of cell cultures using phase contrast microscopy and multivariate imaging techniques*, Stats and Control Meeting, May 23-24 2012, Hamilton (McMaster University).
- **Juneau P.M.**, Garnier A., Duchesne C., *Using phase contrast microscopy to study cells growth kinetics and morphology*, Quebec-Ontario Biotech meeting, May 30-31 2012, Ottawa (Université d'Ottawa).
- **Juneau P.M.**, Duchesne C., Garnier A., *New Analytical Algorithms to Characterize Cells Kinetics and Morphology Using Phase Contrast Microscopy*, 62nd Canadian Chemical Engineering Conference, Vancouver, Canada, 2012.
- Juneau, P.M., Duchesne C., Garnier A,, *The Undecimated Wavelet Transform-MIA (UWT-MIA) for Texture Segmentation*, 2nd International Workshop on Multivariate Image Analysis, Valencia, Spain, May 23-24, 2013 (poster presentation).
- **Garnier A.**, Duchesne C., Juneau P.M., *Analysis of the Growth Kinetics and the Morphology of Human Muscle Precursor Cells (hMPCs) using Image Processing and Multivariate Techniques*, 63rd Canadian Chemical Engineering Conference, Fredericton, Canada, 2013.
- Juneau P.M., Garnier A., Duchesne C., *USING PHASE CONTRAST MICROSCOPY TO CHARACTERIZE THE MORPHOLOGY AND THE GROWTH KINETICS OF HUMAN MUSCLE PRECURSOR CELLS (hMPCs) IN SEVERAL CULTURE MEDIA VIA IMAGE PROCESSING AND MULTIVARIATE TECHNIQUES*, 11th meeting of the International Society for Stem Cell Research, June 12 - 15 2013, Boston (MA), USA (poster presentation).

- Juneau P.M., Garnier A., Duchesne C., *Characterization of the morphology and the growth kinetics of human muscle precursor cells (hMPCs) in several culture media via image processing and multivariate techniques*, 2013 Till & McCulloch meetings, October 22 – 25 2013, Banff, Canada (poster presentation).
- Juneau, P.M., Duchesne C., Garnier A., *Using Gabor Wavelet Features and Multivariate Image Analysis Techniques to Assess the Impact of Culture Medium on Myoblast Morphology Observed Under Phase Contrast Microscopy*, Cell Culture Engineering (CCEXIV), Quebec City, Canada, May 4-9, 2014 (poster presentation).
- Juneau, P.M., **Duchesne C.**, Garnier A., The Undecimated Wavelet Transform Multivariate Image Analysis (UWT-MIA) for Texture Segmentation, EuroPact 2014, Barcelona, Spain, May 6-9, 2014 (presentation performed by Dr. Duchesne).
- **Juneau P.M.**, Garnier A, Duchesne C., *Using the UWT-MIA method to characterize myoblasts morphology*, Stats and Control Meeting, May 22-23 2014, Montreal (Polytechnique).

Chapter 1. Introduction

The study of how cells react to different culture conditions is facilitated by the use of microscopy to record the cellular responses over an extended period of time (5-10 days), generating for each experiment over 100 GB of image data consisting of about 40,000 frames, with up to thousands of cells in each frame (Kang, Miller et al. 2006). Automated cell detection and tracking is important in many research fields such as wound healing, embryo development, immune system studies, cancer research, parasite spreading, tissue engineering, stem cell research and drug research and testing (Paduano, Sepe et al. 2010). Studies of *in vitro* cellular behavior typically involve thousands of cells, and interpreting these vast amounts of data via manual analysis is time-consuming, costly and prone to human error (Theriault, Walker et al. 2012). In response to the need for automated analysis tools, machine vision methods were developed to detect and track cells in time-lapse microscopy, but cell segmentation remains an extremely challenging task (Theriault, Walker et al. 2012). In cell-based research, an accurate quantification of cell numbers is important for further analysis, and before the advent of high-throughput screening (HTS) manual cell counting in each image and the use of a hemacytometer were standard methods (Polzer, Haasters et al. 2010). However manual counting is too time consuming and tiring in HTS applications and the use of the hemacytometer requires trypsinization for adherent cells, thus being quite intrusive and providing only rough estimates of cell numbers (Polzer, Haasters et al. 2010). As a consequence, for microscopy-based HTS applications, there is a great need for automated cell counting and image processing methods that are robust and reliable (Polzer, Haasters et al. 2010).

In order to optimize the formulation of culture media for growing *in vitro* animal or human cells, HTS is becoming a widely accepted methodology since it can test a huge variety of molecule combinations (growth factors, cytokines, etc.) at different concentrations using design of experiments (DOE) techniques. In order to assess the impact of culture conditions on the growth of animal cells, HTS platforms increasingly rely on microscopy, since it is possible to extract critical information from image processing such as the degree of confluence (cell-covered surface), cell counts, cellular morphology, cell-cell interations and provide information about how cells grow, move or even undergo differentiation. Because of the huge quantity of data and images that can be generated even by a single HTS experiment, image mining techniques (Berlage 2005; Berlage 2007) are required,

involving as much automation as possible for image processing. In fact, user interations is of the main bottleneck in high-throughput imaging applications, and therefore the development of more reliable and automated image analysis techniques for cellular imaging remains an important goal in computational molecular biology and microscopy-based HTS (Cheng and Rajapakse 2009).

From an automation and robustness perspective, fluorescence microscopy has several advantages in microscopy-based HTS such as the possibility to target specific molecular markers with fluorescent probes (e.g. immunofluorescence) and also it makes image processing for single-cell segmentation and tracking easier and reliable since it yields higher signal to noise ratio images compared to brightfield or phase contrast microscopy (Ali, Gooding et al. 2007). For segmenting cells stained with a fluorophore, a common and simple approach is to first binarize the image using a thresholding technique and then use the watershed transform on either intensities, gradients, shapes, other measures or a combination of all of those (Padfield, Rittscher et al. 2011). However, fluorescence microscopy for HTS has also several drawbacks. For instance, adding antibodies, dyes or fluorophore probes into a cell culture involves procedures that are time consuming and is an intrusive process that can affect cellular biology mainly due to cytotoxicity (Polzer, Haasters et al. 2010). Also, fluorescence microscopy can induce cellular damage due to UV exposition (Frigault, Lacoste et al. 2009), which can be detrimental for the cells, not mentioning that the residual labeling molecules can make it difficult to use the same cells afterwards for cell therapy. As a consequence, there is a need to develop non-intrusive HTS methods relying on microscopy that could provide as much information as possible about the cells. These drawbacks explain why the present project aims at developing imaging techniques relying on phase contrast microscopy (PCM), which are less intrusive.

Phase-contrast microscopy (PCM) is a well-established tool to image living cells in biomaterials engineering research (Theriault, Walker et al. 2012). PCM has the advantage of being non-invasive to study cells that are sensitive to phototoxicity (Xiong, Chia et al. 2011). There is still a need to develop image processing methods to analyze PCM images, since only a small number of commercially available softwares is capable of quantifying cells, and these softwares are often elaborate, costly, customized for specific applications or often require profound programming skills that are not accessible for every researcher (Polzer, Haasters et al. 2010). Image mining and processing algorithms comprise a large

array of methods and tools that can be freely or commercially available, and these softwares need to balance between being too open-source (only experienced programmers can operate them) or too closed-source (too rigid in their usage or too expensive to develop and maintain) (Berlage 2005). Even if there is a growing number of reports about image processing algorithms for PCM images, it is difficult to compare them directly because each algorithm is designed for different situations, including different imaging modalities, microorganisms or cell types, and benchmark datasets (especially for PCM). Ways to evaluate different techniques on a common basis are not readily available (Theriault, Walker et al. 2012).

Key issues still remain to be addressed in image analysis algorithm development (especially for PCM images) and this area is still described as one of the greatest remaining challenges in screening applications (Carpenter, Jones et al. 2006). The bottleneck today is putting and analyzing together images, image-derived information and metadata from experimental databases to allow for the analysis of global patterns and to distinguish and evaluate the information at different levels (pixel level, segmented object level, culture conditions level and pattern level) (Berlage 2007). While existing softwares aim at analyzing particular assays for particular cell types, high throughput image analysis has mainly been impractical unless an image analysis expert developed a customized solution or unless a commercial package was used with its built-in algorithms applicable for a limited set of cellular features or for a limited set of cell phenotypes (Carpenter, Jones et al. 2006). There still exists a clear need for powerful and flexible open-source platforms for high-throughput cell image analysis (Carpenter, Jones et al. 2006). Many research groups developed automated cell counting and characterization platforms, but most of them rely on high-contrast images produced in fluorescence using labeling or are tailored for specific applications and require elaborate programming skills inaccessible for many researchers (Polzer, Haasters et al. 2010). Also, the varying experimental approaches reported in the literature demand considerable ressources (in terms of tuning) for setting-up the segmentation parameters and for performing features selection and occurences classification (Berlage 2005).

As a consequence, the aim of this project was to develop robust and reliable image processing algorithms for PCM-based high-throughput screening applications. To develop these image processing techniques, human myoblasts were imaged in live-cell microscopy

(in PCM, so it is not intrusive), studying the impact of culture time, passage number and culture media on the cells. The images acquired during this project were used to develop and test the algorithms and image processing approaches proposed (summarized in three articles). The techniques developed during this project are robust enough so they can be applied on a large database of PCM images, providing information about cellular growth kinetics and cellular morphology. The image processing approaches developed through this project are robust against uneven illumination and noise in order to ensure that reliable information is obtained from PCM (Kenong, Gauthier et al. 1995), making it possible to detect overall patterns related to culture conditions and cellular phenotypes. Even though the algorithms were developed for myoblasts, they may be used and tuned for other applications involving adherent cell lines such as endothelial cells (Hoesli, Garnier et al. 2014).

For perfoming HTS experiments relying on microscopy, a good knowledge of cell culture (particularly of the cell line studied), microscopy and image processing methods is required. Once the images are acquired and processed, multivariate techniques are required to extract pertinent information from a large number of image features to highlight the best (or the worst) culture conditions. The literature review presented in Chapter 2 focuses mainly on these four items (cell culture, microscopy, image processing and multivariate analysis), which are important background knowledge in every microscopy-based HTS project.

The aim of this PhD project was to develop image processing algorithms to analyze live-cell images acquired in phase contrast microscopy in order to extract valuable data. The ultimate goal is to be able to use (in the future) these algorithms in high-throughput screening applications involving live-cell imaging. As a basis for this project, a human myoblast cell line (provided by the lab of Dr. J.P. Tremblay, CRCHUL) was used to test the algorithms developed in this PhD project, as it was known that the cells morphology is influenced by the culture medium used. The methodology used to acquire microscopic images to build a database (relative to the culture media used and the passage number) is provided in Chapter 3. The algorithms developed were published in several papers. A simple algorithm using a range filter, a threshold and a pixels size filter was tuned to estimate the cell-covered surface in each image in order to assess the myoblasts growth kinetics through passages (Chapter 4). To relate the texture patterns within PCM images

to the cells morphology, a new MIA algorithm relying on the continuous undecimated wavelet transform (UWT-MIA) was developed (Chapter 5). The UWT-MIA was then tested on the whole PCM image database to establish a statistical link between the culture medium used and the morphology of the cells (Chapter 6). These algorithms provide the ability to estimate cell counts and analyze the morphology of adherent cells grown on flat surfaces in PCM without performing single cell segmentation, and could be beneficial for future high-throughput screening applications. A link between cellular morphology and cell type could be investigated in future work, and, if confirmed, the image processing techniques developed could be used to assess cellular functionality in real time.

Chapter 2. Literature Review

2.1 High throughput Screening (HTS)

High throughput screening (HTS) or high content screening (HCS) is directed at evaluating the response of experimental pharmaceuticals or cytotoxins on large numbers of increasingly small cell samples arranged in microarrays (Freshney 2010). HTS refers more generally to the quantification of processes that are defined spatially and temporally in the array of cells (Berlage 2005). The main advantage of a HTS system is that they can assay simultaneously a very large number of samples with relatively few cells while recording and reducing data instantaneously with minimal manual intervention (e.g. via image processing) (Freshney 2010).

A good overview of the whole process behind a high throughput screening experiment was presented by Rimon (2011), and is presented in Figure 2-1. The first step is to define the biological question to answer. Then, depending on the system to study, the appropriate labeling methods and HTS platforms are identified. Once the screening is completed, the images (or other forms of data) are processed using different softwares performing data extraction and classification (machine learning or object classifiers). Quality control is usually performed using false positives and false negatives samples that need to be recognized by automatic data processing softwares. If quality control criteria are not met, image and data processing steps need to be repeated in order to have robust results. Data analysis allows biological hypotheses to be formulated, confirmed or infirmed via statistical analysis. Finally, to help the scientific community comparing results (and the performance of the image processing and data analysis algorithms), data should be shared and made available to an accessible database.

As technology evolves, highly-performant high throughput screening platforms appeared on the market, providing the ability to analyze in an automated fashion a large quantity of samples. However, as pointed out by Rimon (Rimon and Schuldiner 2011), there is a danger that as it becomes easier to process large quantities of samples to produce very large datasets, it becomes also harder to process all this data to obtain valuable biological information and conclusions. Sydney Brenner referred to this phenomenon by saying that high throughput experiments are in danger of creating “low-input, high-throughput, no-output biology” (Brenner 2008). Even though high throughput screening platforms are

maturing, a major goal remains to develop high throughput screening technologies that will be “systematic, accurate, fast and unbiased without giving up the requirement to provide profound and highly informative data” (Rimon and Schuldiner 2011). One of the main challenges is to process all the data in a robust manner to be able to draw significant and reliable scientific conclusions.

Another main challenge is to be as little intrusive as possible when performing HTS experiments to make sure that additives do not influence too much cellular behaviors, as already mentioned (Polzer, Haasters et al. 2010).

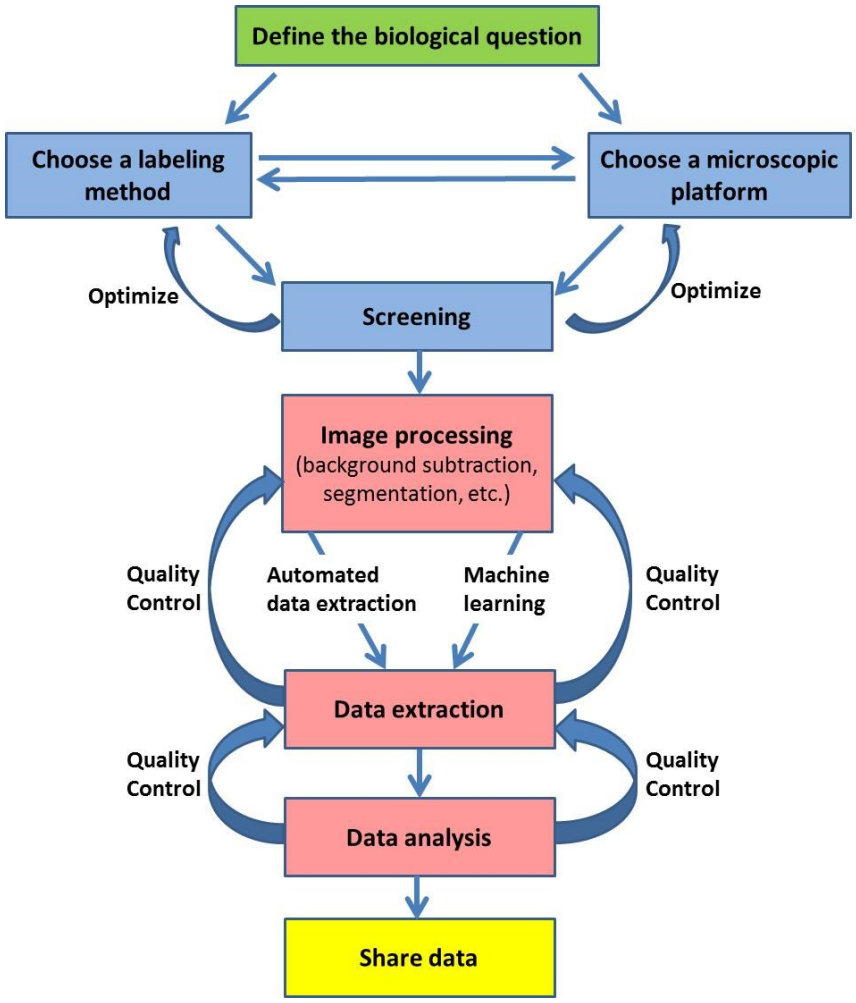


Figure 2-1 : Whole process behind high throughput screening (HTS) experiments (adapted from Rimon and Schuldiner (2011)).

The labeling methods (to highlight cellular structures of interest) and the experimental strategies (HTS platforms) chosen will influence the type and quality of information that can be obtained (whole organism or single cell structure and behavior), as illustrated in Figure 2-2. It is thus very important to consider all the aspects of the biological questions to solve in order to identify the appropriate experimental set-up.

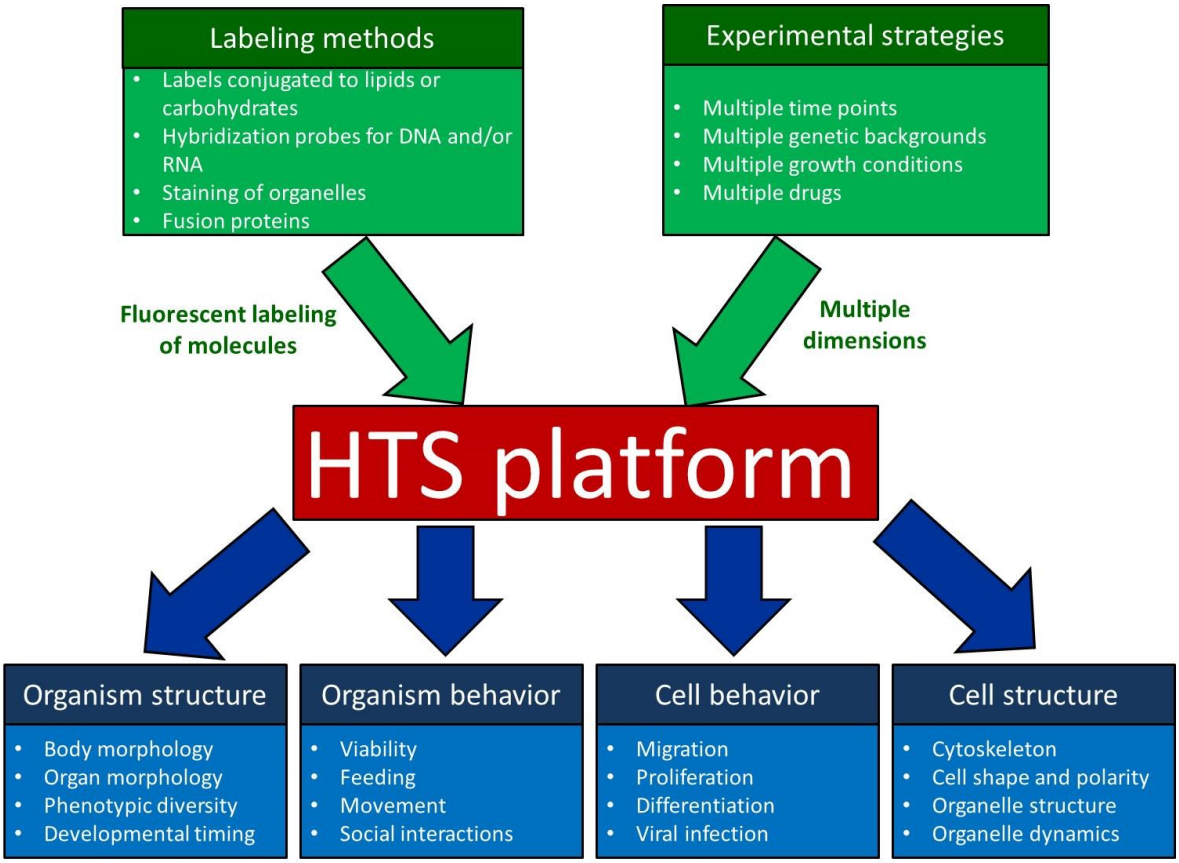


Figure 2-2 : Influence of labeling methods and experimental strategies on the type and quality of information that can be obtained on whole organism or single cell structure and behavior (adapted from Rimon and Schuldiner (2011)).

For HTS applications, the most common labeling methods are given in Table 2-1.

Table 2-1 : Labeling methods in HTS (Rimon and Schuldiner 2011)

Cell component of interest	Labeling method
Ions	Chemical sensors Genetically encoded sensors
Sugars	Fluorescent analogs
Lipids	Lipids dyes Anti-lipid antibodies Lipid-binding fluorescent fusion proteins (for example: a PH domain fused to GFP)
Proteins	Antibodies Fluorescent fusion proteins
DNA, RNA	Probes DNA- and/or RNA-binding proteins Promoter activity by fluorescent reporters
Organelles	Antibodies Fluorescent fusion proteins Lipid dyes Bright-field microscopy
Whole cells	Membrane lipid dyes Bright-field microscopy Antibodies

HTS platforms relying on microscopy use different types of optical set-ups to obtain the information required to draw conclusions via image processing. Usually data extracted from image processing needs to be processed by machine-learning algorithms (or classifiers). A good review paper on how to use image features and machine-learning methods to distinguish cellular phenotypes in HTS was written by Sommer et al. (2013). The objective of a HTS experiment is to verify if a perturbation (e.g. treatment with a drug, small interfering RNA or genetic manipulation) would lead to a new cellular phenotype which is characterized by a change in cellular morphology, protein expression level or anything that can be measured by imaging biosensors (Sommer and Gerlich 2013). More and more HTS applications relying on microscopy and image processing are reported in the literature. For instance, an automated microscopy HTS set-up was used as a drug screening platform to identify molecules capable of fighting resistant microbes such as *Staphylococcus aureus* (Rajamuthiah, Fuchs et al. 2014), to quantify changes in the cytoskeleton of cancerous cells which are submitted to different drugs (Vindin, Bischof et al. 2014) and to identify cellular morphologies that are specific to a certain gene knockdown (Failmezger, Frohlich et al. 2013). Using microscopy-based HTS platforms requires background knowledge in cell culture, microscopy (applied to biological samples), image processing and multivariate analysis. A literature review on these four subjects is provided in the next sections.

2.2 Culture of myoblasts

2.2.1 Cell culture

Muscle precursor (satellite) cells and myoblasts are eukaryotic cells that belong to the myogenic lineage. They play an important role in myogenesis, the process of muscle generation and repair, in which myoblasts multiply, differenciate and ultimately undergo fusion to actual or new muscle fibers (see Section 2.2.3). Myoblasts can be cultivated in order to treat by cell therapy muscular dystrophy, a medical condition in which the patients skeletal muscles weaken and progressively disappear because of the absence of a functional protein named dystrophin (Hoffman, Brown et al. 1987). In cell therapy, healthy myoblasts are cultivated *in vitro* and transplanted into dystrophic patients in order to fuse to their muscle fibers and express dystrophin (Skuk, Goulet et al. 2007).

Myoblasts can be grown *in vitro* onto a support, which is particularly suitable for HTS experiments since it is possible to control the pH, the temperature, the osmotic pressure,

the culture medium and the O_2 and CO_2 partial pressures (Freshney 2000, p.4). The efficiency of different supports for the culture of myoblasts as well as different fusion inhibitors were already tested in the literature (Boudreault, Tremblay et al. 2001).

Myoblasts used for HTS and research applications are generally seeded in a multiwell plate or a T-flask and are incubated for about 4-5 days until they are confluent (meaning that the adherent cells cover most of the surface available). The advantage of using multiwell plates for proliferation tests is to be able to test in parallel different culture conditions (such as different culture media), providing the opportunity (using an automated microscope) to image each well (and several locations within each well) through time in a HTS fashion. Using 6 or 12 wells multiwell plates is preferable, since they decrease the variability in the cell count results compared to 24 or 96 wells multiwell plates. Also, 6 or 12 wells multiwell plates are easier to image in phase contrast microscopy, since it is easier to maintain phase contrast on a larger area (lower meniscus effect).

Since the culture medium used has an important impact on cellular growth and functionality, providing essential nutrients and critical chemical messengers to the cells, it is important to replace it frequently (medium change). Myoblasts growth is influenced by cytokines in the culture medium, which are factors that are released by cells that will induce receptor-mediated effects on the proliferation, differentiation or inflammation of other cells (Freshney 2010). Growth factors present in the culture medium will also enable a chemical communication between cells and influence their cell cycle via signalling pathways. However several cytokines have a short lifespan in culture medium at the incubation temperature, and thus the culture medium is usually changed every 2-3 days. For example, the concentration of Fibroblast Growth Factor 2 (FGF-2) can decline as much as 80% after 24 hours at 37 ^{0}C without heparin (Caldwell, Garcion et al. 2004). According to Freshney (Freshney 2000), the factors that would indicate that the culture medium should be changed are a drop in pH and a change in cell morphology (a deteriorating morphology is usually a sign that it is time to change the culture medium). The cell concentration and the type of cells will influence the rate of change of the culture medium (Freshney 2000).

Once the cells reach about 80% confluence, it is preferable to harvest them via trypsinization so they do not differentiate furthermore and form myotubes. Trypsinization is the process in which a proteolytic enzyme (trypsin) is added to the cell culture to break down the proteins that attach the cells to their substrate (e.g. T-flask surface) (Freshney

2010). Figure 2-3 shows typical images of myoblasts imaged at 10X in phase contrast microscopy at low (A) and high (B) confluence as well as undergoing fusion to form myotubes (C):

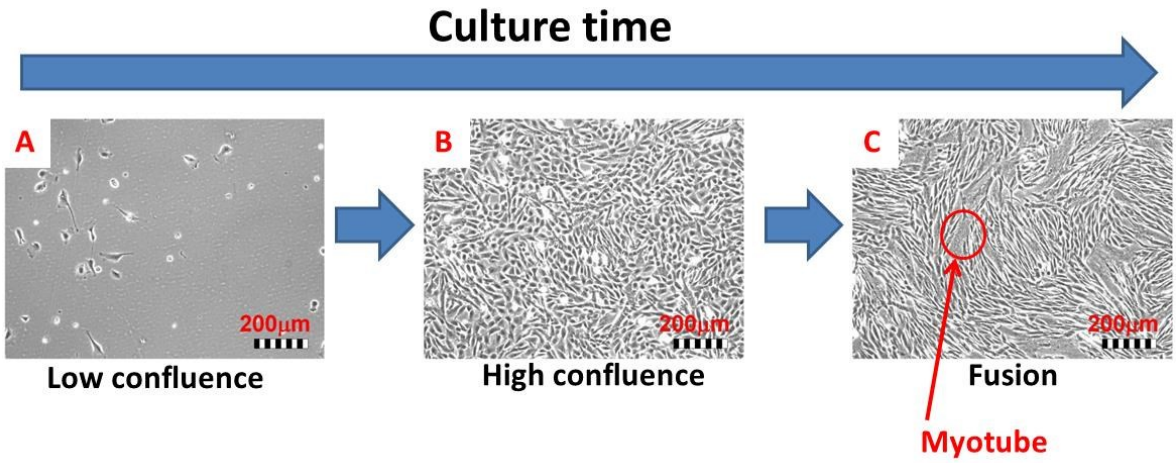


Figure 2-3 : Typical images of myoblasts at low (A) and high (B) confluence, as well as undergoing fusion (C) at a magnification of 10X

A passage is achieved when the cells are put back into culture (at a lower concentration) after being trypsinized. Trypsinization is performed at the end of a cell culture (at confluence) to harvest cells. The cells obtained from a fresh biopsy can be cultivated and expanded over several passages, as shown in Figure 2-4.

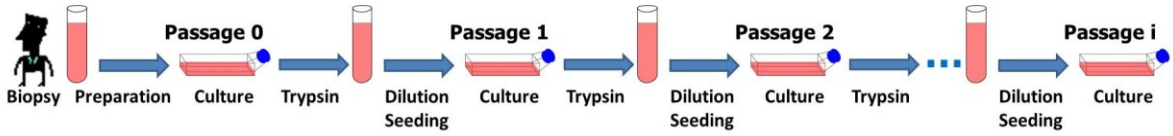


Figure 2-4 : Illustration of how cells from a biopsy can be cultivated over several passages

The number of passages performed (related to the age and the number of generations associated to the cells) has a non-negligeable effect on the performance of the cell cultures: higher doubling times, less growing cells, decrease in cellular density, enhanced contact inhibition, difficulty to adhere to a solid substrate and a decrease of the response to growth factors. According to Freshney (Freshney 2000), for a particular cell line, it is possible to grow the cells over several passages up to a point where they go into senescence (maximum number of divisions programmed genetically). This phenomenon

would be due to the incapacity of the terminal DNA sequences of telomeres to replicate after each division, and the result is a progressive shortening of the telomeres up to a point that the cells are not capable of dividing anymore (Freshney 2000). Also, trypsinization can alter the cells metabolism and can be unfavorable to human stem cells as unnecessary passaging must be avoided to preserve the cells plasticity (Polzer, Haasters et al. 2010).

2.2.2 Culture medium

Most cell lines are grown in a serum-supplemented medium, in which a serum (animal blood extract from which cells and coagulation proteins are removed) is added to a basal medium (containing glucose, salts, amino acids, etc.) and several additives. The most popular sera for tissue culture are fetal bovine serum (FBS) and calf serum (CS). Serum contains growth factors, proteins, and hormones which promote cell proliferation and cell adhesion (Freshney 2010). Since it is a biological extract, its exact compositon (in terms of growth factors, lipids, etc.) is not known and varies for each lot. As a consequence, using serum for cell culture has several drawbacks such as physiological variability, shelf life limitations, quality control issues and potential contamination problems (Freshney 2010). These issues explain why important research efforts are made for developing and testing serum-free media for cell culture, for which the composition is known and can be controlled (fully defined media).

For the culture of human myoblasts, three culture media were used in this project: a serum-supplemented medium (SSM), a serum-free medium (SFM) and a differentiation medium (DFM).

As described by Parent (2009), a standard SSM medium for the culture of myoblasts uses MCDB120 as a basal medium (see Table C-4 for composition) supplemented with FBS and a few additives (SSM composition provided in Table C-1).

Since several growth factors and cytokines can favor muscle cells mitosis, it is possible to replace FBS to elaborate a serum-free medium (Deasy, Jankowski et al. 2001). A patent for a serum-free medium (SFM) called LOBSFM was submitted by Parent et al. (2009). This medium can sustain the growth of human myoblasts as efficiently as a SSM over 60 days, using different growth factors, cytokines and additives. The LOBSFM (referred to as SFM) composition is provided in Table C-2.

When a cell culture reaches a high degree of confluence (80%-90%), one can replace the culture medium by a differentiation medium (DFM) to induce myoblast fusion (formation of myotubes) (Parent 2009). A typical DFM is constituted of a standard medium (RMPI, F12 or MCDB120) with ITS (1X) (Parent 2009). It is possible to add 2% of horse serum (HS) to favor fusion (Parent 2009). A DFM medium was tested on human myoblasts by Parent (Parent 2009), which can induce the formation of myotubes in 2 to 3 days. Another option is to use the Wnt signalling pathway to promote myotubes formation by using a culture medium supplemented with human recombinant Wnt-3A (rh-Wnt3A) (Agley, Velloso et al. 2012).

According to Parent (2009), even if the cells grown in SFM express less desmin in their cytoskeleton than the cells grown in SSM, they keep their capacity to differentiate and to undergo fusion. The culture medium used has an influence on surface markers and receptors associated to myogenic cells (see Section 2.2.3.1). However, even if myoblasts grown in SFM lose desmin and have a decreasing number of NCAM receptors with time, they keep their capacity to form myotubes after six weeks of culture (Parent 2009).

2.2.3 Cellular phenotypes associated to the myogenic lineage

Myogenesis is a complex process in which muscles are generated through several distinct phases during the development of the embryo and in the adult organism (Bentzinger, Wang et al. 2012). In the perinatal phase, muscle resident myogenic progenitors first proliferate and their growth rate decreases as the number of myonuclei (nuclei of a muscle fiber (Shenkman, Turtikova et al. 2010)) reaches steady state and myofibrillar protein synthesis peaks (Bentzinger, Wang et al. 2012). When the muscle is mature, progenitors become quiescent and reside within the muscle as satellite cells (Bentzinger, Wang et al. 2012). Adult skeletal muscles constitute renewable organs that rely on homeostasis mechanisms to compensate for cellular turnover, and their myogenesis depends on the activation of satellite cells so they can differentiate into new myofibers (Charge and Rudnicki 2004; Bentzinger, Wang et al. 2012). The myogenesis process associated with the repair of damaged mature muscle tissue is relatively well known: large numbers of satellite cells expand and differentiate to repair the tissue and re-establish homeostatis (Rudnicki, Le Grand et al. 2008; Bentzinger, Wang et al. 2012). Figure 2-5 shows how myofibers in a typical skeletal muscle are surrounded by satellite cells.

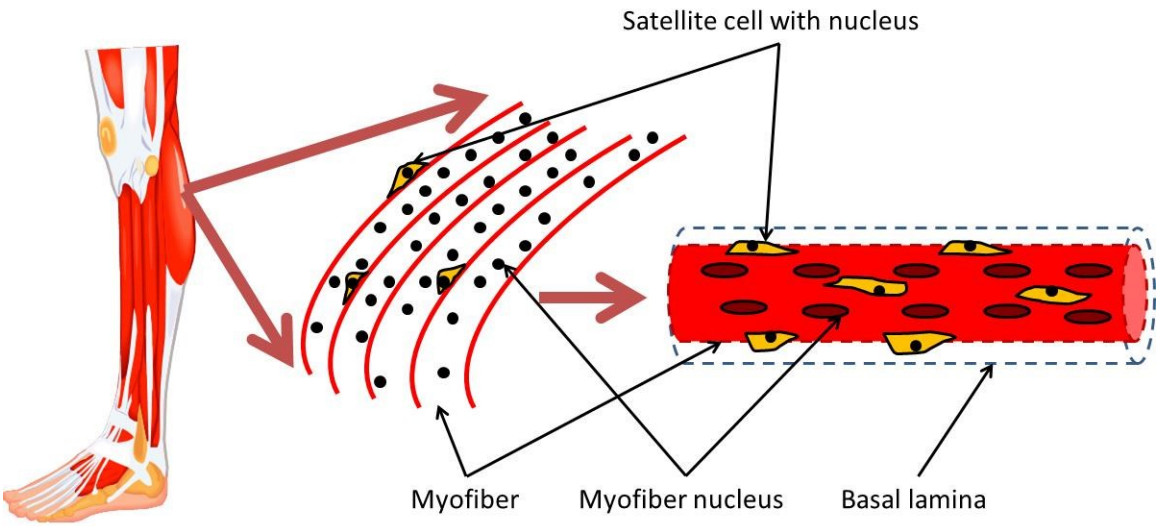


Figure 2-5 : Typical skeletal muscle with myofibers surrounded by satellite cells (adapted from Charge et al. (2004) and Danoviz et al. (2012))

Different cellular phenotypes are involved in myogenesis: stem cells (embryonic progenitors), satellite stem cells, satellite committed cells, proliferating myoblasts, quiescent myoblasts (myocytes) and myofibers/myotubes (Charge and Rudnicki 2004; Bentzinger, Wang et al. 2012).(Bentzinger, Wang et al. 2012) For the purpose of the current project, only myogenic lineage phenotypes are considered (shown in Figure 2-6).

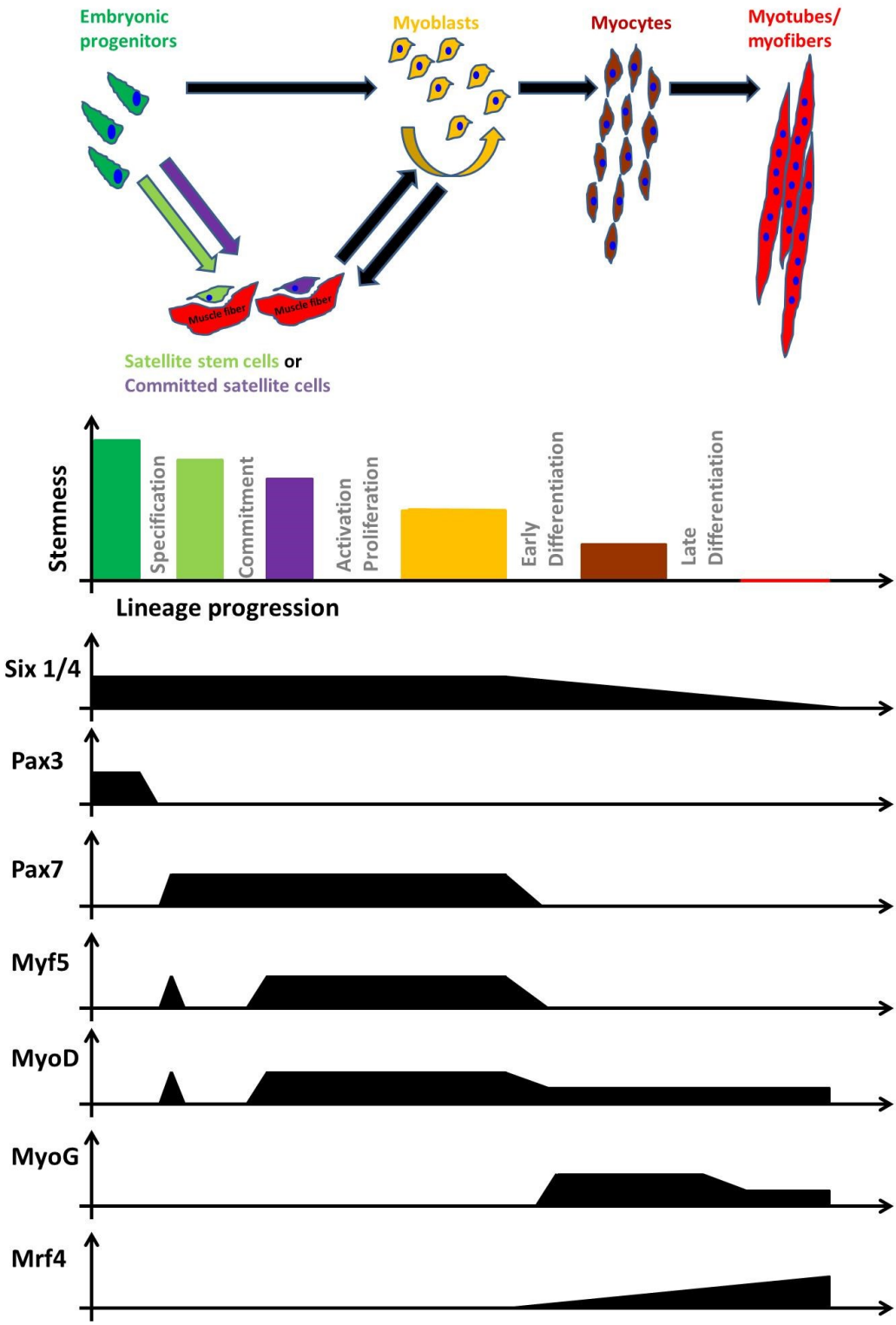


Figure 2-6 : Different phenotypes in the myogenic lineage as well as relevant molecular markers (adapted from Bentzinger et al. (2012))

Stem cells can generate myoblasts or satellite cells. Myoblasts (positive for MyoD and/or Myf5) are muscular cells that can get out of their cell cycle to become quiescent. Once the myoblasts become quiescent (myocytes), they can then fuse and form multinuclear cells called myotubes or myofibers. Satellite cells can generate proliferating myoblasts. However satellite cells are difficult to cultivate, since they have a strong tendancy to differentiate rapidly into myoblasts (Montarras, Morgan et al. 2005; Bentzinger, Wang et al. 2012).

Satellite stem cells and commited satellite cells form a niche. The satellite stem cell polarity and spindle orientation relative to the basal lamina determines whether the stem cell division will be symmetric or asymmetric (Rudnicki, Le Grand et al. 2008). Planar divisions (parallel to the basal lamina) are symmetrical, generating identical daughter stem cells (Rudnicki, Le Grand et al. 2008). In contrast, apical-basal divisions (90^0 relative to the basal lamina) are asymmetrical, with one daughter cell remaining a satellite stem cell at the basal surface and a commited satellite daughter cell destined for differentiation on the apical surface (Rudnicki, Le Grand et al. 2008). It is possible to distinguish different myogenic cellular phenotypes by relying on several molecular markers and sometimes by looking at cellular morphology.

2.2.3.1 Molecular markers

Several molecular markers (cell surface, cytoskeletal and transcription factors markers) can be used to distinguish myogenic cellular phenotypes. A rough list of important molecular markers (associated genetic factors) is provided by Buckingham, Bajard et al. (2003).

Table 2-2 : List of molecular markers (genetic factors) associated to different stages of skeletal muscle formation (Buckingham, Bajard et al. 2003)

Stage	Associated genetic factors
Delamination	Pax3, c-met
Migration	c-met/HGF, Lbx1
Proliferation	Pax3, c-met, Mox2, Msx1, Six, Myf5, MyoD
Determination	Myf5, MyoD
Differentiation	Myogenin, Mcf2, Six, MyoD, Myf6
Specific muscle formation	Lbx1, Mox2
Satellite cells	Pax7

For example, the molecular markers listed in Table 2-3 are expressed by proliferative and quiescent satellite cells (Charge and Rudnicki 2004).

Table 2-3 : Molecular markers for satellite cells (Charge and Rudnicki 2004)

Molecular Markers	Quiescent Satellite Cell	Proliferative Satellite Cell
Cell surface		
M-cadherin	+/-	+
Syndecan-3	+	+
Syndecan-4	+	+
c-met	+	+
VCAM-1	+	+
NCAM	+	+
Glycoprotein Leu-19	+	+
CD34	+/-	+/-
Cytoskeletal		
Desmin	-	+
Transcription factors		
Pax7	+	+
Myf5	+/-	+
MyoD	-	+
MNF	+	+
MSTN	+	-/+
IRF-2	+	+
Msx1	+	-

Deasy and Jankowski (Deasy, Jankowski et al. 2001) present an overview of the different molecular markers of interest providing information on the stem cells as they undergo differentiation to generate myofibers. Figure 2-7 shows the molecular markers associated to the whole differentiation process (Deasy, Jankowski et al. 2001).

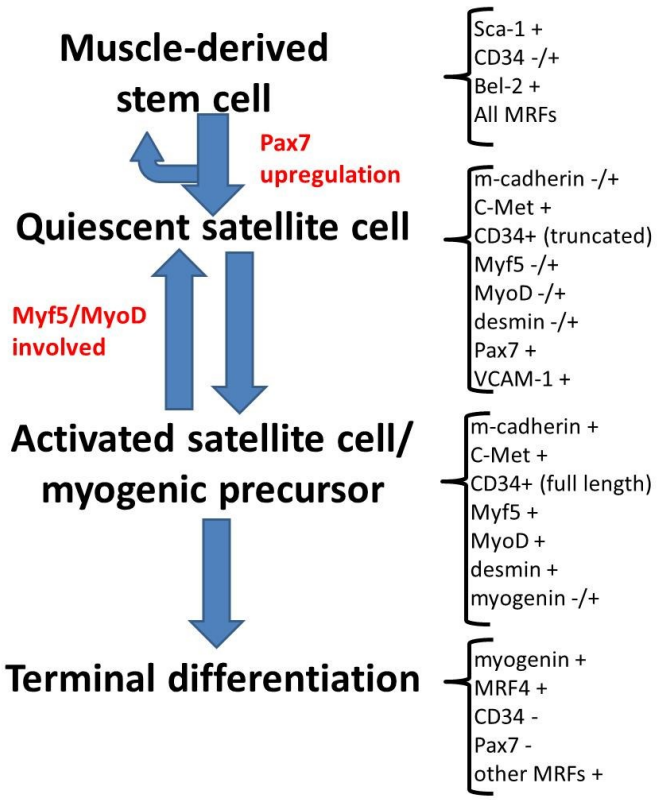


Figure 2-7 : Molecular markers associated to the differentiation of stem cells to muscle cells (adapted from Deasy et al. (2001))

The expression of several molecular markers as the cells undergo differentiation in the myogenic lineage was illustrated by Bentzinger et al. (Bentzinger, Wang et al. 2012), and is shown in Figure 2-6. MyoD, Myf5, Myogenin and MRF4 (also known as Myf6) are regulating genes collectively expressed in the skeletal muscle lineage and are referred to as the myogenic regulatory factors or MRFs (Bentzinger, Wang et al. 2012). Several progenitors remain as satellite cells in postnatal muscles and form a heterogeneous population of stem and commited satellite cells (Bentzinger, Wang et al. 2012). Activated committed satellite cells (myoblasts) can return to the quiescent state when the muscle is regenerated (Bentzinger, Wang et al. 2012). Six 1/4 and Pax 3/7 regulate the early lineage specification; Myf5 and MyoD commit cells to the myogenic phenotype as the expression of the terminal differentiation genes (when the myocytes fuse to form myotubes) are performed by both MyoG (myogenin) and MRF4 (Bentzinger, Wang et al. 2012).

To characterise a cellular lineage and different cellular phenotypes, immunofluorescence assays or ELISA techniques can be used (Freshney 2000). These techniques rely on

antibodies specific to the molecular markers of interest. The most popular marker to differenciate myoblasts from other types of cells is the desmin protein (cytoskeleton protein). After fixing the cells with ethanol (95%), it is possible to incubate the cells with a desmin antibody, to rince and to incubate again with a secondary antibody tagged with a fluorophore such as Alexa 488 or Alexa 546 (Parent 2009). Usually the cells nuclei are stained with DAPI to be counted. It is recommended to perform the test on at least 500 cells to have a representative sample of the population's phenotype. Several surface markers such as NCAM and MHC (Myosin Heavy Chain) can be used to detect myoblasts via flow cytometry (Parent 2009).

2.2.3.2 Morphology

According to Freshney (2000), myogenic cells (myoblasts) proliferate and migrate in a random fashion on a substrate and align themselves as they reach confluence. Myoblasts then undergo a fusion (final differentiation) process to form multinucleated cells (myotubes). It is possible to cultivate myogenic cells over several passages by a series of trypsinization/seeding, but once cells undergo final differentiation (fusion), their proliferation is difficult to estimate.

This change in cellular morphology as myogenic cells undergo differentiation is illustrated roughly by Bentzinger et al. (2012) (see top of Figure 2-7) and also by Rudnicki et al. (2008) in Figure 2-8, where small round mononucleated embryonic progenitors can differentiate and produce long multinucleated myotubes, a process in which transcription factors have an important role. It is possible to separate satellite cells from myotubes using flow cytometry (by differentiating cells as they are Pax3 positive or negative) when processing muscular tissue (Montarras, Morgan et al. 2005). The separated cells can be plated afterwards to observe their morphology (Montarras, Morgan et al. 2005). This means that with the proper molecular markers (detected by antibodies/fluorescent markers), it is possible to separate the different myogenic phenotypes and to plate them to check their morphology.

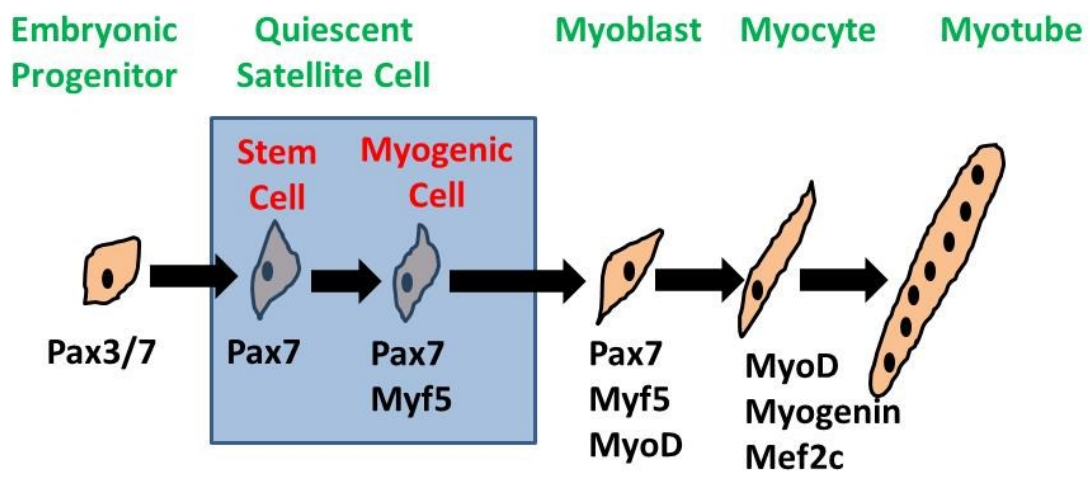


Figure 2-8 : Rough description of the cellular morphology through the myogenic lineage (adapted from Rudnicki et al. (2008))

However, it is important to note that other factors such as culture conditions will influence cellular morphology and myogenic differentiation. For example, the differentiation of the mouse myoblast cell line C2C12 can be accelerated by a high gravitational force *in vitro* (Nakajima, Hashimoto et al. 2011): the myoblasts thicken and have a different morphology. The differentiation of myoblasts to myofibers was verified by applying periodical electrical impulses to observe contractive movements.

The culture medium used can influence (via its cytokines, growth factors and hormones) the cellular metabolism, growth kinetics and phenotype of myogenic cells and thus their morphology. For example, myoblasts grown in SFM (LOBSFM) contain more granules, form more vacuoles and are more elongated (more delicate) than myoblasts grown in SSM, showing a certain degree of cellular deterioration. Also, when inducing cellular fusion, freshly isolated muscle precursor cells (MPCs) expanded and differentiated in the presence of rh-Wnt3A formed larger myotubes (with a greater average size) than untreated controlled MPCs (Agley, Velloso et al. 2012).

2.2.4 Cell morphology

2.2.4.1 Morphology characterization

Cells respond to environmental signals that regulate their behavior (cell differentiation, division and migration), and scientists and engineers study changes in their morphology to understand the influence of those signals (Theriault, Walker et al. 2012), involving

sometimes image processing for cell segmentation. The behavior of vascular smooth muscle cells and fibroblasts is interesting to observe in order to investigate cell-substrate interactions *in vitro* (Theriault, Walker et al. 2012). The morphology of adherent cells can be roughly assessed qualitatively according to their spread, polarization and orientation, as illustrated in Figure 2-9 (Theriault, Walker et al. 2012). Just after seeding, cells are defined as “non-spread” and are characterized by a small round area surrounded by a high contrast halo due to the reflection of the incident light on their spherical membrane (Theriault, Walker et al. 2012). This appearance is also typical of cells undergoing division or apoptosis (Theriault, Walker et al. 2012). The transition from the “non-spread” to the “spread” state involves the polymerization of the cytoskeleton elements to push outward the cell membrane and increase the area of the cells attached to the substrate (Theriault, Walker et al. 2012). After the initial spreading, the cells can either be “polarized” or “non-polarized”: non-polarized cells have a smooth round membrane whereas polarized cells have extended protusions (pseudopods) supported by their internal cytoskeleton (Theriault, Walker et al. 2012). Cells in the polarized state can be classified has being oriented or not: non-oriented cells have protusions evenly distributed around the cells as stongly polarized cells show a significant level of bilateral symmetry (Theriault, Walker et al. 2012).

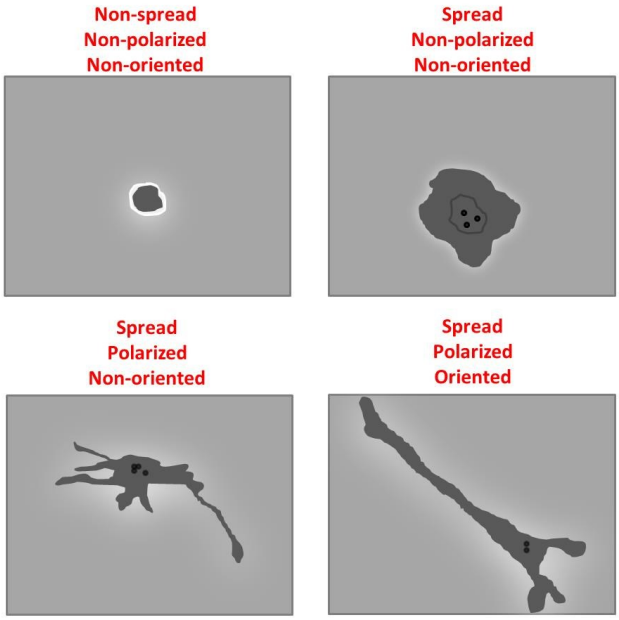


Figure 2-9 : Qualitative characterization of the morphology of adherent cells in PCM (adapted from Theriault et al. (2012))

However, since the cells shape changes continuously, they can go through partial states (partially spread, polarized or oriented), and thus intermediate labels should be considered (Theriault, Walker et al. 2012).

Quantitative mathematical models were also developed to assess cellular morphology. For example, a top-down rule-based mathematical model to explore the basic principles related to mechanochemical events involved in animal cell migration was proposed by Satulovsky et al. (2008). In their work, the cells were modeled as a shape machine that protrudes or retracts in response to a combination of local protusion and global retraction signals (depending of what is in the culture medium), and an optimization algorithm was used to identify the parameters that generate specific shapes and migration patterns. The authors were able to account for the behavior of *Dictyostelium* under a large collection of conditions, imaging the cells in PCM with a magnification of 40X (Satulovsky, Lui et al. 2008). It appeared that motile activities in different regions of a cell collectively determine the cell shape, and that these are linked to cellular growth and viability (Satulovsky, Lui et al. 2008). The shape machine model is illustrated in Figure 2-10 (Satulovsky, Lui et al. 2008).

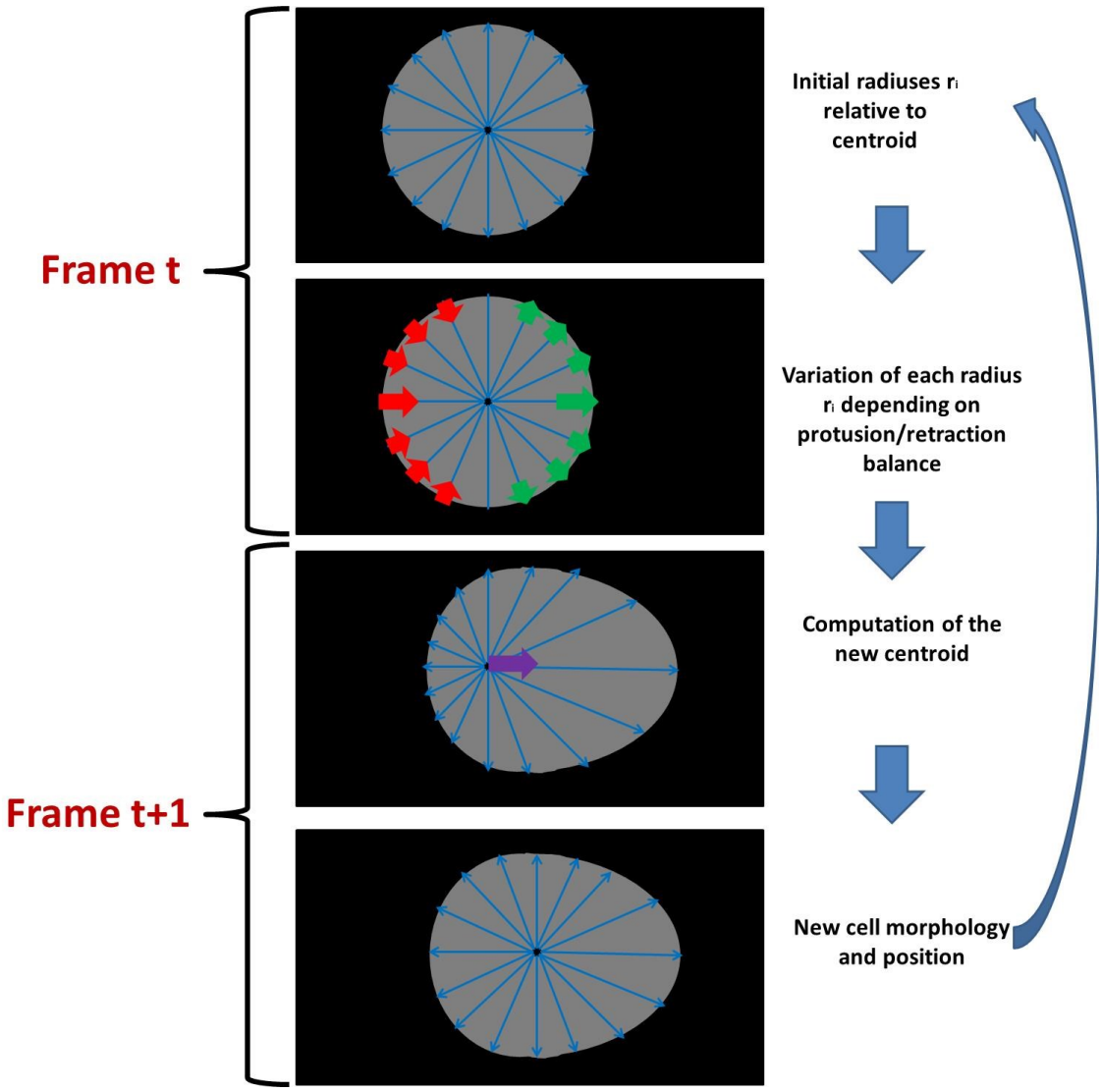


Figure 2-10 : Shape machine model for cellular morphology (adapted from Satulovsky et al. (2008))

A cell is modeled by its perimeter points and its center defined as its geometrical centroid at a given frame. The distance between the perimeter points and the centroid (x_c, y_c) corresponds to the radiuses r_i expressed in polar coordinates (θ_i, r_i), with *i* ranging from 1^0 to 360^0 (roughly one point per degree) (Satulovsky, Lui et al. 2008). For each frame or time point, the balance between the local protusion signal and the global retraction signal is performed, and as each radius varies in length, the cell will move. The cell can move its perimeter points away from the center by localized protusions (in green) or towards the center by retraction (in red), and the centroid position can be updated accordingly (Satulovsky, Lui et al. 2008). Each radius length will change depending on the protusion

signal (collective signalling activities stimulating actin polymerization near the membrane) and the retraction signal (long range signals that stimulate myosin-dependent contractility). The balance between the protusion and retraction signals (modelled by differential equations with specific parameters) will determine the evolution of the points at the cell perimeter (Satulovsky, Lui et al. 2008). It is then possible to characterize the shape and migration of the cells with several metrics such as the maximal radius, actual area, area fluctuations, speed, roundness and persistence (Satulovsky, Lui et al. 2008). It is important to note that this model deals with the cell shape and migration in two dimensions, and it is unnecessary to consider the conservation of the total volume of the cells, assuming that the cells will change their height in the third dimension. The movement of the centroid as a result of perimeter extension and retraction implies adhesive interactions with the substrate and thus implies energy and work not taken into account in the model (Satulovsky, Lui et al. 2008).

Most quantitative models describing cellular shape and motility use a large number of experimental parameters coming from the literature (protein concentration, kinetic constants, rheological moduli of the cytoplasm, etc.). In the model proposed by Satulovsky et al. (2008), underlying mechanochemical events are implicitly taken into account in the mathematical rules and constaints, with each parameter representing a “lumping” of many molecular interations. Such a model aims at being combined with experimental manipulations to determine the function of specific molecules or structures in cell migration and morphology (Satulovsky, Lui et al. 2008).

2.2.4.2 Morphology and cellular state

Changes in cell shape (morphological changes) constitute a macroscopic manifestation of different intracellular molecular processes such as the actin-polymerization pressure at the cell edge, myosin-dependent contraction of the actin network, the adhesion to an extracellular matrix and membrane tension (Ambuhl, Brepsant et al. 2012), thus providing insights into the cellular state. Cellular morphology is one of the fundamental criteria to visually classify cells into histologic types, since there is a strong relationship between the shape of the cell and its functional integrity in several specialized cells such as neurons (Olson, Larson et al. 1980). Aspect of shape maintenance in cells has been linked to growth control in normal cells, and several observations suggest that the shape of adherent cells changes during viral and oncogenic transformations (Olson, Larson et al. 1980).

Because of the link that exists between cell morphology and state, it can be hypothesised that cell shape is also related to cell health. According to Freshney (2000), it is vital to observe the morphology of the cells during a cell culture do detect any deterioration, such as granules around the nucleus, vacuoles in the cytoplasm or an increase in roundness that would be due to the fact that the cells detach from the substrate. Such deteriorations in cellular morphology can be due to a problem with the culture medium, a contamination issue or related to cellular senescence, and capturing these potential problems as early as possible is important since they are often irreversible (Freshney 2000).

From a modelling perspective, cellular shape dynamics can be simulated via a mathematical model based on specific hypotheses about the internal dynamics and the state of the cells (as illustrated by Satulovsky et al. (2008)). Cellular shape simulations can be compared to experimentally observed cell shapes measured by live-cell imaging using segmentation and tracking computation algorithms to analyse microscopic images (Ambuhl, Brepsant et al. 2012). These simulations can link cellular state to cellular morphology from a mathematical perspective.

2.2.4.3 Measuring the morphology of myogenic lineage cells

The accurate measurement of the morphological characteristics of adherent cells with nonuniform shapes (such as cells of the myogenic lineage) is difficult (Agley, Velloso et al. 2012). Myoblasts usually generate pseudopods, which are transient protusions of the cellular membrane used for locomotion and phagocytosis which are important to detect correctly via image processing when performing mobility and morphology studies (Zimmer, Labruyere et al. 2002). Because pseudopods are usually partly or entirely localized outside of the focal plane, they appear at a much lower contrast than most of the cell membrane in PCM, and thus tend to be treated by snakes (active contours) as occlusions and are ignored (Zimmer, Labruyere et al. 2002). Also the correct detection and analysis of myotubes in microscopic images is associated with several challenges which make their size analysis difficult, as described in Table 2-4 (adapted from Agley et al. (2012)). A traditional method for measuring myotube size is to calculate the average myotube diameter derived from multiple measurements along their length (Agley, Velloso et al. 2012). However this method cannot deal with the often irregular conformations of these cells *in vitro* (Agley, Velloso et al. 2012). A common method to characterize the degree of differentiation of myogenic cells is the fusion index, which is the number of nuclei inside

myotubes as a percentage of the total number of nuclei in the entire field of view (Boudreault, Tremblay et al. 2001; Agley, Velloso et al. 2012) counted using a stain (DAPI or Hoechst).

Table 2-4 : Challenges related with the size analysis of myotubes in microscopic images (Agley, Velloso et al. 2012)

Challenges	Implications for myotubes size analysis
Many myotubes are non-cylindrical	Diameter measurements would be roughly inaccurate.
Large variability in myotube sizes	Averaging myotube sizes may conceal an experimental finding at a particular point within the size range.
Nuclear clustering	It is difficult or impossible to calculate a fusion index and nuclear incorporation.
Presence of mononuclear cells in myotube cultures	High-throughput analysis based on proteins associated with terminal differentiation (e.g., MHC) may pick up myoblasts, which express also these markers (i.e. it is not specific).
Formation is highly dependent on seeding density	High plating densities improve the statistical likelihood of myoblasts differentiation, fusion and myotube formation. However this may overwhelm and conceal the effect of an experimental agent on fusion. The inverse might be true for low seeding densities.
Whole myotubes often extend beyond limits of image field	Many experimenters may be reporting the measurement of myotubes when they are in fact measuring segments of myotubes or “myosegments”.
Experimenter bias in manual image processing	Techniques such as diameter measurements, manual tracing of myotubes, and selection of “small” or “large” myotubes are all open to bias.

When examining epifluorescence microscopic images, Agley et al. (2012) suggested the following criteria to identify single human myotube or myosegment:

1. Cellular entities with more than two nuclei;
2. Cellular entities that express MHC;
3. Polynucleated structures satisfying points 1 and 2 joined by any common cytoplasmic region (with no separating boundaries) are considered as a single myotube or myosegment;
4. If the entire outer membrane is visible within the image, the cell is defined as a myotube;
5. If the entire cell body/outer membrane is not visible within the image, the cell is classified as a myosegment.

The common method to image myoblasts and myotubes is to use molecular markers and to take epifluorescence or confocal microscopic images. For example, using epifluorescence microscopy (immunocytochemistry) with proper fluorescent markers, it was possible to quantify the cytoplasm and nuclei surface of myotubes (stained with Hoechst for nuclei and marked for MHC for their cytoplasm) as well as the cytoplasmic, nuclear and myonuclear areas of myoblasts (stained with Hoechst for nuclei, marked for desmin for their cytoplasm and marked for Ki67 to highlight specific areas within the nuclei associated to proliferative cells) (Agley, Velloso et al. 2012). It was then possible to use different shape features (via image processing) to characterize their morphology (Agley, Velloso et al. 2012).

2.2.5 Cellular growth

2.2.5.1 Introduction

For most microorganisms, growth is an essential response to their physical, chemical and nutritional environment (Shuler and Kargi 1992). In a proper culture medium, microorganisms such as animal cells will extract nutrients from their environment and will use them for biosynthesis, product formation and energy production (Shuler and Kargi 1992). As a consequence, cellular growth can be considered as an autocatalytic process (Shuler and Kargi 1992) in which the number of cells will increase over time due to their own replication. The increase of cell number over time in a batch set-up (most of the time animal cells are cultivated in T-flasks or multiwell plates) can be represented by a growth curve generated by manually counting cells at specific time intervals, by image analysis of

the growing culture or by colorimetric tests (Freshney 2010). Growth curves can be analyzed further by fitting kinetic models with varying degree of complexity and number of parameters to quantify and predict the behavior of the cells according to their environment.

The degree of precision of the growth kinetic parameters estimated highly depends on the cell counting methods used. There are mainly two classes of methods to determine cellular growth: direct and indirect methods (Freshney 2010):

- Among direct methods, hemacytometer counting and the use of a Coulter counter are the most popular. A hemacytometer is a glass slide with a small chamber having a known volume which is used to manually count cells from a culture sample. The cells are counted over a specified surface and are translated back into an estimate of the cellular concentration. A Coulter counter is a machine that processes a cell culture sample and estimates the cell concentration by measuring fluctuations in electrical resistance (Freshney 2010). Image processing methods relying on single-cell segmentation in microscopic images (via image processing) can also be considered as a direct method.

- Among indirect methods, several platforms can correlate cellular concentration to the mitochondrial activity (XTT), to the total DNA quantity (CyQuant) or to the DNA synthesis rate via the incorporation of [^{3}H] thymidine (Parent 2009; Freshney 2010).

Several factors influence growth kinetics. Cytokines, oxygen and nutrients have an important impact on growth rates, and should be at some point incorporated into kinetic models (Tabatabai, Bursac et al. 2011). Also, the initial seeding concentration used at the beginning of each culture can impact cellular growth (Brinkmann, Lutkemeyer et al. 2002). For myoblasts coming from a biopsy, the number of passages can have an impact on growth kinetics (Juneau, Garnier et al. 2013).

When using brighfield or phase contrast microscopy images to assess growth kinetics of adherent cells, it can be really difficult to segment them individually. It is possible to use the cell-covered surface (degree of confluence) instead (He, Wang et al. 2007), assuming that the number of cells is roughly proportional to the cell-covered surface (Topman, Sharabani-Yosef et al. 2011).

2.2.5.2 Kinetic growth models

There are two types of kinetic growth models: models that consider that a cell population can grow without any limitations (unbounded models) and models that consider that the cell population can't grow over a limited carrying capacity (bounded models). Usually it is possible to fit an unbounded model at low cell concentrations (low confluency) and to fit a bounded model on a whole range of cell concentrations.

2.2.5.2.1 Unbounded models

At the beginning of a cell culture, there can be a lag phase followed by an exponential growth phase (Shuler and Kargi 1992). The biomass X (usually expressed in cells/mL) varies through time as a function of a specific growth rate μ (in h^{-1}). Usually, at low cell concentrations (low confluence), cells have an exponential growth phase that can be modeled by a first order kinetic model (Shuler and Kargi 1992):

$$\frac{dX}{dt} = \mu X \qquad (2\text{-}1)$$

With the initial boundary condition $X=X_0$ at t=0, it is possible to integrate the previous equation to obtain the following equation (Shuler and Kargi 1992):

$$X = X_0 e^{\mu t} \qquad (2\text{-}2)$$

The doubling time (DT) of a cell culture is defined as the time required to double the number of cells. It is expressed relative to μ as follows (Shuler and Kargi 1992):

$$DT = \frac{\ln 2}{\mu} = \frac{0.693}{\mu} \qquad (2\text{-}3)$$

Another interesting unbounded model is the Sherley model that accounts for non-dividing cells (Deasy, Jankowski et al. 2003):

$$X = X_0 \left[0.5 + \frac{1-(2\propto)^{\frac{t}{DT}+1}}{2(1-2\propto)}\right] \qquad (2\text{-}4)$$

Where α is the mitotic fraction, DT is the cell division time and X_0 is the initial cell concentration. A kinetic model proposed by Deasy and Jankowski (2003) for stem cells includes terms for proliferative (mononucleated) and non-proliferative differentiated (polynucleated) cells, taking into account the heterogeneity of the myogenic population. This model takes also into account the mortality of the cells (variable M):

$$X^{\text{Mononucleated}} + X^{\text{Polynucleated}} = X_0 \left[0.5 + \frac{1-(2\propto)^{\frac{t}{DT}+1}}{2(1-2\propto)}\right] - M \quad (2\text{-}5)$$

This model was used to fit data from time-lapse microscopy live cell imaging with images taken every 10 minutes over 5 days (in phase contrast). The number and the nature of the cells were assessed by lineage trees. Cells were counted using Hoechst labeling and their proliferative nature was assessed by immunostaining (Deasy, Jankowski et al. 2003). They have also shown that there is an increase in the cells cycle time when a confluency higher than 50% is reached, explaining why the cells behave overall under a logistic model (Deasy, Jankowski et al. 2003). A weakness of the Sherley and Deasy models is that they don't consider a maximum carrying capacity (growth is unbounded, like the exponential model) (Tabatabai, Bursac et al. 2011).

2.2.5.2.2 Bounded models

Among all models describing a sigmoidal growth curve, it is possible to consider the logistic model, the Gompertz model, the Richards model, the hyperbolastic models (Tabatabai, Bursac et al. 2011) and the Michaelis-Menten model (Shuler and Kargi 1992). All those models use a maximum carrying capacity (X_{max}) and require an initial cell concentration (X_0 at $t_0 = 0$) to be solved. In the case of adherent cells, the space available on the substrate for cellular growth is limited, and a bounded model considers the maximum support capacity of the culture environment by incorporating a parameter for the maximum cell quantity achievable (X_{max}). If the culture is monitored from the beginning to the end, a logistic model can be used to represent the cell concentration as a function of time (Shuler and Kargi 1992):

$$\frac{dX}{dt} = C_{max} X \left(1 - \frac{X}{X_{max}}\right) \quad (2\text{-}6)$$

For most animal adherent cells, there is some form of contact inhibition since, as the cells grow and reach a higher density on the substrate, they form clusters and only the cells at the periphery would have the space required to undergo mitosis. A kinetic model incorporating contact inhibition in the specific growth rate μ was proposed by Lim and Davies (1990), in which C and μ_{max} are kinetic parameters:

$$\frac{dX}{dt} = \mu X \quad (2\text{-}7)$$

$$\mu = \mu_{max}\left[1 - \exp\left(-C\left(\frac{X_{max}-X}{X_{max}}\right)\right)\right] \quad (2\text{-}8)$$

To represent the complexity of stem cells growth, more advanced mathematical models (involving more parameters) are also proposed in the literature, such as the Richards, the Gompertz, the Weibull and the hyperbolastic models (Tabatabai, Williams et al. 2005; Tabatabai, Bursac et al. 2011). Assuming that both the surface available for the cells and the substrate in the culture medium are limited, the Michaelis-Menten model (Shuler and Kargi 1992) can be considered and is expressed as follows (assuming that substrate consumption is directly proportional to the number of cells):

$$\frac{dX}{dt} = \frac{\mu_{max}(X_{max}-X)X}{K_S * Y_{X/S} + (X_{max}-X)} \quad (2\text{-}9)$$

Where μ_{max} is the maximum growth rate when there are no limitations and $K_S*Y_{X/S}$ is a parameter representing the negative impact of substrate consumption and surface availability on cellular growth.

2.2.5.3 Myogenic cell growth kinetics

For primary human myoblasts, Boudreault et al. (2001) reported a μ_{max} between 0.57 and 0.78 d^{-1} using a serum-supplemented medium (SSM), depending on the surface/volume ratio used for cell cultures. For the LOBSFM culture medium (SFM), a DT of 1.05 days, corresponding to a μ of 0.66 d^{-1}, was reported (Parent 2009), meaning that both SSM and SFM media used for this PhD project induce similar myoblasts growth kinetics (in terms of cell counts).

2.3 Microscopy

2.3.1 Types of microscopy

According to Freshney (2000, p.37), an inverted microscope is an essential tool to observe a cell culture regularly, providing a mean to detect different problems such as a change in cells morphology or even contamination. A microscope can also provide the opportunity to observe cells when testing different conditions and performing high throughput screening experiments.

A microscope is by definition an apparatus relying on a set of lenses to obtain a magnification of an observed specimen (Cox 2007). A lens is a piece of glass that redirects parallel light rays towards a precise point location (focus). The distance between

the focus and the lens is called focal distance. The magnification effect comes from the fact that the smaller the focal distance is, the bigger the real image obtained is. In a microscope, the utilization of a second lens (two lenses in series) to redirect the light towards the eye creates a virtual image which is much bigger than the object, as illustrated in Figure 2-11:

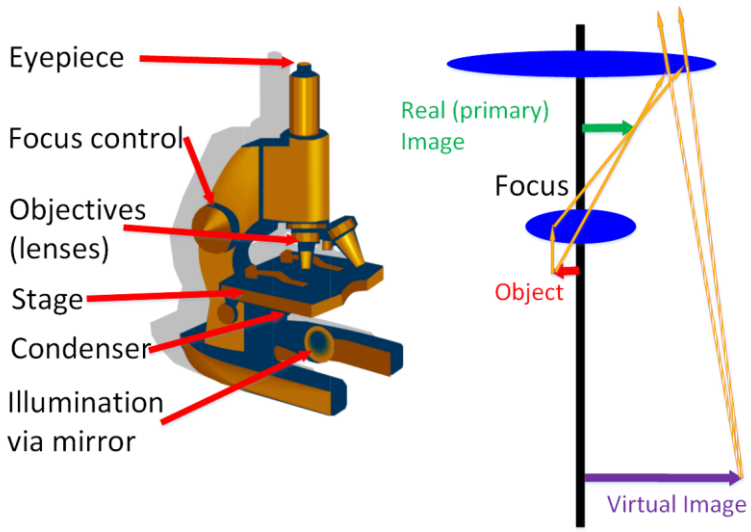


Figure 2-11 : A second lens to redirect the light towards the eye of the observer creates a virtual image much bigger than the original object (adapted from Cox (2007))

A short description of the different types of microscopes for studying biological samples is provided in Table 2-5. It is possible to find more details in Cox (2007). Since phase contrast microscopy (PCM) and hyperspectral microscopy (HM) are of particular interest for this thesis, they are described in more details in dedicated sections.

Table 2-5 : Short description of the different types of microscopes

Brightfield Microscopy (BF)
• Koehler illumination: a lens is placed directly in front of the lamp (lamp condenser), ensuring a uniform illumination at the focal point of the lens, for the specimen. • The field iris can adjust the area of the specimen exposed to light. • The condenser redirects the light towards the specimen, and the condenser iris controls the angle at which the light reaches the specimen. • A variant is Quantitative Phase Microscopy (QPM), in which a stack of brightfield images is processed (Curl, Harris et al. 2004; Selinummi, Ruusuvuori et al. 2009).
Darkfield Microscopy (DF)
• Only the light diffracted by the specimen can form an image, and only the structure of the specimen of interest is visible. • The specimen or the structure under study will be illuminated, and the background will stay dark since it does not diffract light. • A ring at the condenser only allows light rays arriving at an angle larger than the acceptation angle of the objective to pass, making sure that no incident light will be conserved.
Phase Contrast Microscopy (PCM)
• The specimen diffracts light and creates a delay of around ¼ λ. • A piece of glass with a ring carved on it is placed at the focal point of the lens of the objective, creating an additional delay of ¼ λ for the diffracted rays. • Globally, the diffracted light rays have a delay of ½ λ compared to incident light rays, creating destructive interference that highlights the structure of the sample (appearing in black on a bright background). • Digital holography microscopy (DHM) can combine phase contrast microscopy and holography to record 3D information of an object (e.g. cells) via interfering wave fronts from a laser (Mõlder, Sebesta et al. 2008).

Differential Interference Contrast Microscopy (DIC)
• In front of the light source, a polarizer is used to polarize the incident light. • Just before the condenser, a Wollaston cube divides the incident light beam into two light beams and a second Wollaston cube placed behind the objective recombines the two polarized light beams. • An analyzer is used to bring both light beams back to the same polarization, thus creating interference patterns. • Between the two polarized light beams, there is already a phase delay of ¼ λ because of a "¼ λ plate". • Biological specimens induce an additional phase shift of ¼ λ for incoming light, providing an overall phase shift of ½ λ (destructive interference). • DIC microscopy is commonly used to observe thick specimens such as protozoa and small organisms. • Similar to quantitative phase microscopy, 4 dimensions (4D) DIC microscopy is an automated system that can record DIC images at multiple focal planes and at multiple time points (Hamahashi, Onami et al. 2005).
Epifluorescence Microscopy (EFM)
• Ploem illumination: the objective serves also as a condenser. • The UV light goes through an excitation filter that only allows specific wavelengths to reach the specimen. • The dichroic mirror reflects short wavelength excitation light and allows longer wavelengths light (emitted by fluorescence) to go to the observer, acting as a beam splitter. • The barrier filter blocks all remaining excitation light before it arrives to the camera or to the observer, and provides the opportunity to distinguish fluorescence generated by different fluorophores.

Confocal Microscopy (CM)
• To image the whole specimen, it is necessary to scan the sample point by point with a moving stage or scanning mirrors. • Light emitted by the laser is reflected by a dichroic mirror towards scanning mirrors to scan the whole specimen through the objective. • Fluorescent light emitted by the specimen is collimated by scanning mirrors and goes through a dichroic mirror. The fluorescent light is then focused in the pinhole and filtered by additional dichroic mirrors to measure light intensity at different wavelengths (different channels). • The main advantage of CM in comparison to EFM resides in the pinhole plate that can exclude fluorescent light emitted outside the focal plane, thus reducing the interference caused by background light emissions, providing higher quality images (Haaland, Jones et al. 2009).
Multiphotons Microscopy (MPM)
• Works in a way similar than CM, at the exception that photons with longer complementary wavelengths are required to excite electrons from the fluorophore instead of a single photon. • Excites the fluorophore with lower energy light, which is easier to perform, avoids photobleaching and specimen degradation. • Fluorophore excitation will occur only at the specimen focal plane (where the laser beams converge) and is less likely to happen higher or lower than the focal plane, thus implying that the use of a pinhole plate is unnecessary. • It is possible to have a detector that acquires a whole spatial image (CCD) instead of a single dot at a time (as in CM).

2.3.1.1 Phase contrast microscopy (PCM)

It was in the 1930s that a dutch physicist (Zernike) came up with the idea to use a phase delay between incident light and the light diffracted by the specimen to create interference to enhance the contrast of images (Cox 2007). Figure 2-12 shows the optical path of the light in a phase contrast microscope.

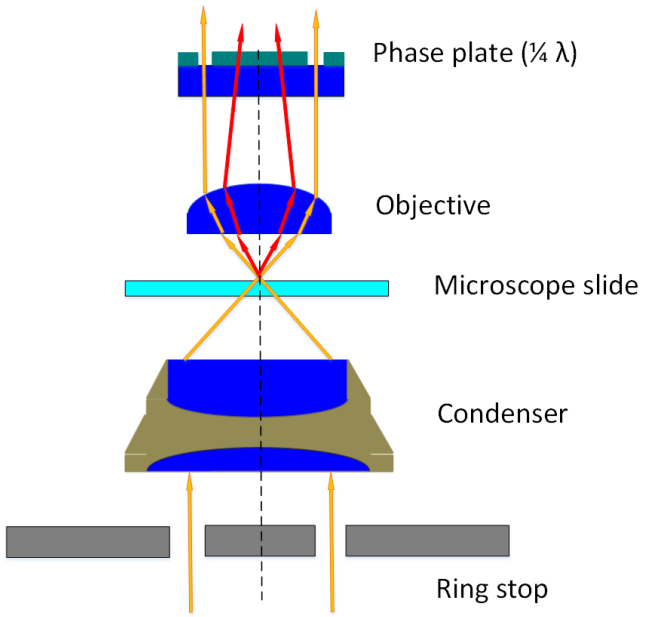


Figure 2-12 : Optical path associated to phase contrast microscopy (adapted from Cox (2007))

A ring is placed at the condenser, allowing only incident light (orange arrows) incoming at a certain angle to illuminate the specimen. The specimen diffracts light (red arrows) and creates a delay of around ¼ λ (depending on the refractive index of the cells). A piece of glass with a ring carved on it is placed at the focal point of the lens of the objective, and this difference in thickness induces an additional delay of ¼ λ for the diffracted rays (red arrows in Figure 2-12) (Cox 2007). Globally, the diffracted light rays have a delay of ½ λ compared to incident light rays, creating destructive interference. This interference highlights the structure of the sample, and the specimen appears in black on a bright background.

Phase contrast microscopy is recommended for thin and uniform specimens such as cellular monolayers (Cox 2007). PCM has the advantage of being a non-intrusive method that can highlight the structure and the shape of the cells, explaining why it was chosen for this PhD project. Phase contrast microscopy is typically used to characterize and track

adherent cells, even though some applications for cellular suspensions have been mentioned in the literature. For example, PCM was used to track suspensions of neural stem cells/neural progenitors using Haar-like features from the wavelet transform output to determine at which focus level these cells where the most likely located (Chao-Hui, Sankaran et al. 2012). Phase contrast microscopy is the preferred source of information when studying cell motility (Ambuhl, Brepsant et al. 2012) and has significant advantages when studying cells morphology and behavior since it provides good contrast for cell edge detection without exogenous dyes, it uses only moderate levels of light and is free of artefacts related with photobleaching and photo-damage that are common in fluorescence microscopy (Ambuhl, Brepsant et al. 2012). From an experimental perspective, if PCM is used in conjunction with fluorescence microscopy, using the phase-contrast channel for edge detection frees an additional fluorescence channel for observing intracellular markers (Ambuhl, Brepsant et al. 2012).

Most phase objects, especially biological cells, have a constant refractive index ranging from n=1.36 to 1.37 at a wavelength of 546 nm (Otaki 2000). The phase difference (or delay) depends on the size of the object. Assuming a spherical object, the maximum phase difference δ and the diameter of the object are related by the following equation (Otaki 2000):

$$\delta = \left(\frac{2\pi}{\lambda}\right)(n' - n)d \qquad (2\text{-}10)$$

Where λ is the wavelength of light in vacuum, n' is the refractive index of the phase object and n is the refractive index of the media. This equation shows that a thicker object (*d* increasing) will create a larger phase difference (δ increasing) for constant refractive indices (Otaki 2000). When imaging cellular monolayers, the thickest portions of the cells will create larger phase differences than thinner portions, as illustrated in Figure 2-13.

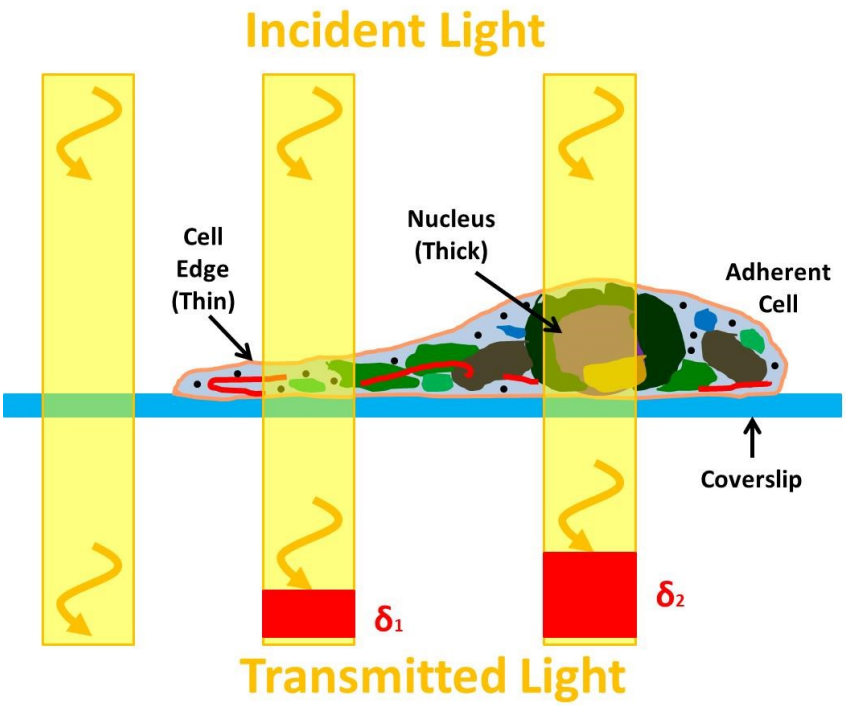


Figure 2-13 : Effect of specimen thickness on phase difference (adapted from Zeiss (2013))

2.3.1.2 Hyperspectral microscopy (HM)

Hyperspectral microscopy is an imaging set-up that provides both spectral and spatial information about a sample under the form of stack of images, called a hyperspectral cube. It is actually the combination of a traditional microscope and a spectrometer (Gao, Kester et al. 2010). Different types of microscopes can be used to image fluorescent cellular specimen (Monici 2005), thus opening the door to couple these microscopes with hyperspectral cameras to capture whole light spectrum. The motivation behind the development of hyperspectral microscopy for biomedical applications comes from the fact that emission and reflection spectra of specimen contain a significant amount of information on the structure, the biochemistry and the physiology of cells. Hyperspectral cubes can be used at the cellular level to distinguish fluorophores that have similar emission spectra or at the tissue level for diagnosis purposes (Gao, Kester et al. 2010).

In comparison to common optical microscopes, hyperspectral microscopy can acquire cubes of data constituted of grey scale images acquired at different wavelengths (Chunni, Qingli et al. 2010). This imaging set-up provides to researchers the possibility of getting

morphological information (spatial) and spectral information related to the biochemistry and the biology of the specimen (Chunni, Qingli et al. 2010).

Most hyperspectral microscopes acquire hyperspectral cubes of data (see Figure 2-19) by scanning the sample in the spatial domain (like a confocal microscope coupled to a hyperspectral camera) or by scanning in the spectral domain by using an acoustic-optic tunable filter (AOTF) or a liquid crystal tunable filter (LCTF) (Gao, Kester et al. 2010). The use of a line spectrometer with a moving stage is also another option, and is called a “push broom” set-up (Chunni, Qingli et al. 2010). In this particular case, a CCD coupled with a diffraction grating provides an image with one spatial and one spectral dimension. It is then necessary to scan linearly the specimen under the camera to provide a second spatial dimension (Katari, Wallack et al. 2009; Polerecky, Bissett et al. 2009; Chunni, Qingli et al. 2010).

A push broom set-up can be used to measure the reflectance (using a Spectralon as a background that reflects incoming light), the transmission or the autofluorescence of a biological sample (Polerecky, Bissett et al. 2009). For example, Polerecky and Bissett (2009) use a line spectrometer, a moving stage and a microscope (magnification of 40X and 100X) to measure the quantity of pigments in biofilm and cyanobacteria samples coming from a coastal region to study benthic ecosystems. A magnification of 100X provides the opportunity to observe algae chloroplasts in fluorescence (Polerecky, Bissett et al. 2009). Katari and Wallack (2009) used an IX-70 Olympus microscope (magnification of 100X) equipped with a line spectrometer and a moving stage (to scan the specimen surface) to detect and distinguish up to 10 different fluorophores in visible light spectrum at the same time. They also modified their apparatus to measure near-infrared spectra (950-1300 nm) to detect lead sulfite quantum dots in fixed mammalian cancer cells (Katari, Wallack et al. 2009).

A new set-up called the Image Mapping Spectrometer (IMS) was developed to perform fluorescence hyperspectral microscopy with high resolution. This set-up is based on the deviation and the dispersion of local zones in the image by a special mirror (Mapping Mirror) and a prism (Gao, Kester et al. 2010). Each pixel on the CCD is associated with a single voxel in the hyperspectral data cube (Gao, Kester et al. 2010). A voxel can be considered as a measurement value (light intensity) in a three-dimensional volume (Cox 2007).

2.3.2 Controlled environment for live cell imaging

2.3.2.1 Chambers on microscope stage

To keep the cells in a healthy environment (at the right temperature and CO_2 concentration), a vast array of environmental chambers exist and a growing number are becoming available (Cox 2007).

In order to perform live-cell imaging, open chambers such as Petri dishes, multiwell plates or T-flasks, can be used, as they are exchanging gases with the atmosphere around them, even though they are sterile. These open chambers are more suitable for small scale experiments/production and high throughput screening applications applied to both adherent and suspension cells. These chambers hold a larger quantity of medium and can maintain their temperature for a long time because the thermal contact with the stage is poor (metal versus plastic), thus providing possibilities for longer live cell imaging experiments (Cox 2007). A major drawback can be the large temperature gradients across the plastic culture dish. Such temperature gradients can cause problems since a variation of as little as 0.5 ^{0}C can cause changes within cells. Without any additional equipment, these chambers are not able to keep a constant temperature and a viable CO_2 concentration for growing cells over several days.

A perfusion (closed) chamber has a continual flow of culture medium over the sample (Cox 2007). The culture medium is pumped from a holding tank kept at a constant temperature and sometimes exposed to gases (oxygen or CO_2), thus providing both temperature and CO_2 control for live cell imaging over several days. Usually the medium is flowed over the sample using a peristaltic pump or by gravity, and it is necessary to verify that no pressure effects are introduced into the chamber because of pulsing (Cox 2007). The drawbacks of a perfusion chamber are the difficulty to avoid bubbles formation and the intricate piping system involved.

2.3.2.2 Microscope enclosure

In order to provide both temperature and CO_2 control for open and perfusion chambers, advanced systems for live cell imaging are using a microscope enclosure (Cox 2007), as illustrated in Figure 3-2 (IX81 Olympus™ microscope with enclosure used for this PhD project). In such an enclosure, it is possible to flow humidified air enriched with CO_2 to maintain partial pressures in the culture medium, and the temperature inside the enclosure can also be controlled using a heating fan that is flowing air across the whole microscope stage and objective (Cox 2007).

Such a set-up can make the access to specific controls of the microscope difficult, but since many microscopes available on the market today are fully automated and can be operated from a computer, this is not a significant issue (Cox 2007). Since a moist atmosphere with high CO_2 concentrations is corrosive and can be prone to fungal and bacterial growth, frequent drying and cleaning is required (Cox 2007). When using open chambers in a humidified CO_2 incubator or in a microscope enclosure, evaporation can lead to an increase of osmolarity in the culture medium, which can influence the growth of mammalian cells drastically (Brinkmann, Lutkemeyer et al. 2002). When using multiwell plates, evaporation in peripheral wells is more significant than in central wells (Brinkmann, Lutkemeyer et al. 2002), and should be considered in experimental designs. For instance, the effect of evaporation on cells growth rate was assessed in 96-wells plates (Brinkmann, Lutkemeyer et al. 2002).

2.3.3 Cell culture sample integrity

2.3.3.1 Non-intrusive analysis

2.3.3.1.1 Visible light

When no additives are added to the cell culture sample, a simple halogen light can be used to study cells with minimal detrimental effects on the cell culture. However, halogen light can have small negative effects such as transferring heat to the cell culture sample because of NIR wavelengths. It is therefore recommended to minimize as much as possible the exposure of the cells to halogen light when performing live cell imaging experiments over long periods.

2.3.3.1.2 Autofluorescence

Autofluorescence is a phenomenon related to the fact that cells produce endogenous fluorophores that can emit low energy light when exposed to UV light (without the need to add anything to the culture). The autofluorescence of certain molecules was investigated by Chance in the last century (Chance, Schoener et al. 1979). Cells contain molecules that can become fluorescent when excited with UV light at the right wavelength (Monici 2005). The majority of the cells autofluorescence comes from their mitochondria and their lysosomes (Monici 2005). The main endogenous fluorophores are the pyridinic coenzyme (NADPH), flavoproteins, aromatic aminoacids and lipo-pigments (Monici 2005). The reduced form of NADH (NADPH) has a maximum excitation wavelength at 340 nm and a maximum emission wavelength at 450 nm (Monici 2005). In the case of flavoproteins, only

the oxidized form is fluorescent with an excitation wavelength between 360 and 450 nm, and an emission wavelength at 520 nm (Monici 2005). According to Monici (2011), the fluorescence associated with these endogenous fluorophores can help quantify the energetic metabolism of cells and tissues. In fact, changes in the status of tissues and cells during physiological and pathological processes can modify the quantity and the distribution of endogenous fluorophores, explaining why autofluorescence can be a good metabolic indicator (Monici 2005). Excitation and emission spectra of endogenous fluorophores associated to autofluorescence are provided by Monici (2005).

According to Sanchez (2010), the degree of autofluorescence of the cells can be used to estimate the quantity of NADPH in breast cancer cells, to distinguish normal, precancerous and cancerous epithelial cells and to define metabolic changes related to apoptosis and necrosis. This phenomenon would be due to the fact that NADH is an important coenzyme for several metabolic processes, including anaerobic glycolysis, the electron transport chain and the citric acid cycle (Sanchez, Prow et al. 2010). The intracellular concentration of NADH (and its distribution) is a natural metabolic indicator of pathological and physiological changes in cells, and the concentration of NADPH (related to the redox potential of each cell) can be estimated by measuring the cells autofluorescence (Sanchez, Prow et al. 2010). Peaks related to autofluorescence can be detected and quantified using multivariate analysis techniques. For instance, using Multivariate Curve Resolution (MCR), Haaland (2009) was able to detect, by exciting at 488 nm, two peaks related to flavoproteins (depending of the state of oxidation).

Several articles report the use of autofluorescence as an indicator of cellular health. Experiments performed *in vitro* and *in vivo* by Monici (2005) on uterine cervical cells showed a significant difference in autofluorescence between normal and cancerous cells. The autofluorescence of keratinocytes cultivated *ex vivo* can be used as an indication of cellular viability, and this measure can be coupled with pH and oxygen consumption data to draw conclusions about cellular health (Sanchez, Prow et al. 2010). The main advantage is to be able to estimate the viability of individual cells by looking at their autofluorescence instead of measuring the viability of the whole cell population. The autofluorescence increases when cells undergo apoptosis, and decreases when they undergo necrosis (Sanchez, Prow et al. 2010). Autofluorescence of tissues in visible and near-infrared domains can be used to have a good contrast between normal tissues and cancerous tissues from kidney biopsies (Demos, Lieber et al. 2005).

The link between autofluorescence intensity and the cell cycle, the cells age and the culture conditions should be explored with caution (Monici 2005). Several authors report that autofluorescence intensity is very low for freshly seeded cell cultures and increases during the exponential growth phase. However, the inverse situation was reported for monolayers of adherent cells and suspended cells, implying that results can vary significantly (Monici 2005). The autofluorescence coming from mitochondria, associated with the respiratory chain, can be affected by metabolic poisons such as rotenone and antimycin A (Monici 2005). These substances are known inhibitors of the electron transport chain, inducing an increase in cellular autofluorescence related to an accumulation of NADPH (Monici 2005).

Even though using autofluorescence to assess cellular health is considered to be a non-intrusive analysis since there are no additives added to cell samples, exposure to UV light can have a negative impact on cell cultures. Monici (2005) considers autofluorescence analysis to be non-destructive, but recognizes that exposure to UV light can cause cellular damages.

2.3.3.2 Intrusive analysis

When using BF, PCM or DIC microscopy, there is often little contrast between the cells and the background, and image processing and analysis remains difficult. Even though autofluorescence could be used (according to the literature) to evaluate cellular health (related with the metabolism), it is highly probable that it won't be enough to distinguish cells phenotypes (myoblasts, myocytes, etc.) and determine the status of the cells in the cell cycle (apoptosis, necrosis, mitosis) only by microscopy. There is often a need to use complementary tools (additives) to distinguish cells from both a chemical and a visual perspective. The use of additives in a cell culture is considered an intrusive analysis, since those chemicals can influence cells behavior and prevent using the cells produced for cell therapy.

2.3.3.2.1 Dyes

Several dyes are available to color the nucleus and the cytoplasm of the cells, creating an additional contrast to distinguish the cells from the background, and also highlighting individual organites. These colorations are generated by the absorption of specific wavelength in the visible spectrum. Eosin and hematoxylin are good examples of dyes commonly used (Siddiqi, Li et al. 2008). Multiple acidic and basic dyes are used in

histology to color cellular structures, and are listed for example in Ross and Pawlina (2006).

2.3.3.2.2 Fluorophores

Fluorescence is a phenomenon in which a molecule (fluorophore) can emit visible light when excited with UV light. A molecule is said to be fluorescent when the absorption of a photon pushes an electron towards a higher energy level (higher orbital)(Cox 2007). This excited electron then reemits a photon when coming back to a lower energy level (lower orbital). However, since a part of the absorbed energy is lost as heat, the emitted photon has less energy that the absorbed photon (higher wavelength)(Cox 2007). A fluorophore is considered to be a fluorescent organic molecule that is used in microscopy. There are several types of fluorophores that can be used in a cell culture.

An important element to consider when using fluorescent molecules is a phenomenon called *photobleaching*, in which a fluorescent molecule will get oxidized over time and will lose its capacity to be fluorescent after a long exposure to UV light (Cox 2007). The propancy of a fluorophore to undergo photobleaching highly depends on the experimental environment, and the exact mechanisms are not well understood, even though several measurement techniques and kinetic models are proposed in the literature (Widengren and Rigler 1996).

Immunolabeling uses antibodies as highly specific probes to mark specific targets on the cells (Cox 2007). In direct labeling, a fluorophore is attached directly to the primary antibody. In indirect labeling, the fluorophore is attached to a secondary antibody which binds to the primary antibody to highlight specific cellular structures (Cox 2007).

Fluorescent stains are fluorescent probes that target specific components or compartments of the cells (Cox 2007). Some probes can be used on living cells, and others can only be used on fixed cells because they do not penetrate within cells or are broken down by them (Cox 2007). Some stains can color the nucleus (DAPI, Hoechst) and several organites such as mitochondria, the endoplasmic reticulum and lysosomes (for example, Mitotracker® for mitochondria) (Cox 2007).

It is possible to genetically modify (by transfection) eukaryotic and prokaryotic cells so they can produce fluorescent proteins, providing the opportunity to observe in real time (through microscopy) several biological processes (Tsien 2009). An excellent literature review on

fluorescent proteins available is provided by Shaner and Tsien (Shaner, Steinbach et al. 2005; Tsien 2009). The fluorescent proteins mentioned are evaluated according to their brightness, their ease of expression, their photostability, their subjectivity to oligomerization, their toxicity and their environmental sensitivity.

2.3.3.2.3 Quantum dots

Quantum dots (or « QDs ») are more and more used as fluorescently labelled inorganic compounds to highlight specific molecules in living systems (Monici 2005). Quantum dots have several advantages over organic fluorophores, since they are brighter, have a good resistance against photobleaching, have wide excitation spectra while having narrow emission spectral bands (Haaland, Jones et al. 2009). All those characteristics, considering also the fact that large emission spectra can be obtained just by playing with the size of the quantum dots, explain why QDs constitute an ideal tool for microscopy involving fluorescence (Haaland, Jones et al. 2009).

2.3.4 Microscopic image acquisition

When using microscopy in general to observe cells in a live cell imaging set-up, several technical aspects related with microscopic image acquisition must be considered in order to get the most relevant information out of each experiment. Amongst all the technical aspects, the degree of magnification (related with resolution), the acquisition grid (related with imaging frequency), the illumination (related with exposure time) and focus adjustment are very important. All those technical aspects are linked together, since a modification in one has an impact on the other. For example, for a given cycle time, increasing the exposure time will likely decrease the maximum imaging frequency achievable. Also, increasing the degree of magnification of the images will increase the resolution in each image, but a larger acquisition grid to scan the same spatial area of the sample will be required.

2.3.4.1 Magnification and resolution

The degree of magnification (and resolution) required depends on what is desired to observe or quantify. For instance, low magnification (5X, 10X) is usually used for cell tracking or to characterize tissues. An intermediate magnification (20X) may be used to observe how cells interact between each other and to have an overall idea of their morphology. High magnification (40X and higher) is used to observe cells internal structures, to track for instance specific proteins inside cells or to track cellular membranes

(Satulovsky, Lui et al. 2008). The degree of magnification is usually controlled by choosing the appropriate objective (4X, 10X, 20X, 40X, 60X, 100X).

Each objective provides a specific resolution (depending of the degree of magnification). The resolution provided by an objective is the minimal distance between two points in an image that can be distinguished. The resolution obtained depends on the wavelength of the incoming light (λ), the acceptance angle of the objective (half angle θ) and the refraction index of the medium (n), as illustrated in the following equation (Cox 2007):

$$R = \frac{\lambda}{2n \sin\theta} = \frac{\lambda}{2NA} \qquad (2\text{-}11)$$

Where R is the resolution (nm) and NA is the numerical aperture of an objective (usually written on one side of the objective).

Most algorithms available for phase contrast microscopy work well for low resolution images, but struggle with high resolution images, thus requiring more development (Ambuhl, Brepsant et al. 2012).

2.3.4.2 Acquisition grid and imaging frequency

When imaging a specimen with a microscope, it is usually impossible to capture the whole sample within one single image at the appropriate resolution. It is thus required to capture several contiguous images in order to scan the surface of the specimen. The tiling pattern used when scanning the specimen is called the acquisition grid and the distance between each field of view captured determines the degree of overlap between images. A sufficient degree of overlap must be used to ensure that no information is lost (oversampling). These concepts are illustrated in Figure 2-14 with the standard Lena image.

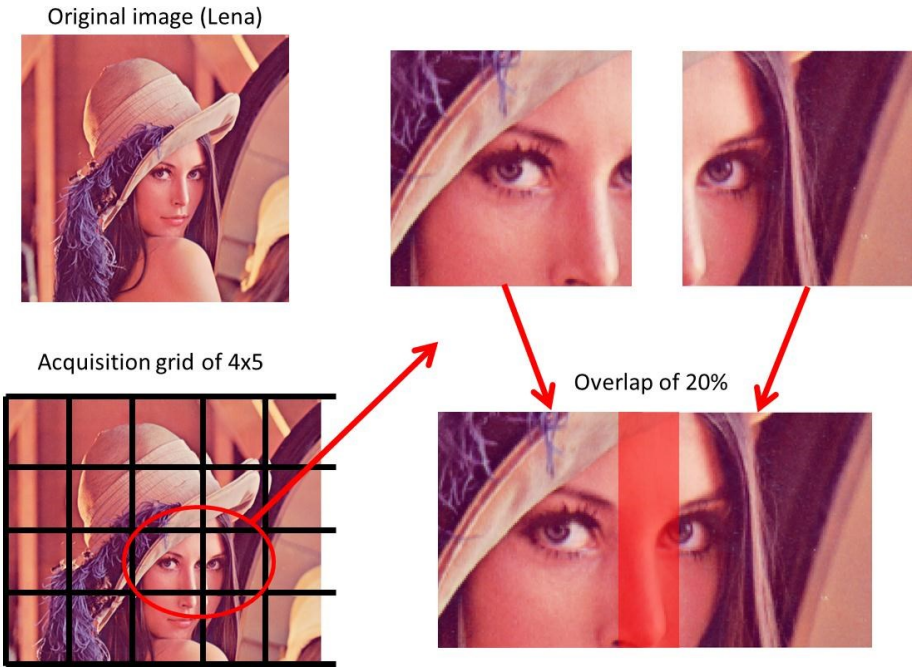


Figure 2-14 : Acquisition grid and overlap concept

Because of the overlap between frames, it is necessary to determine the area common to two adjacent frames and to register them (get rid of information redundancy), as illustrated in Figure 2-14. There are several approaches to image registration. For example, to register images from an acquisition grid and compensate for translation, rotation and scale, Padfield developed an algorithm based on the Fourier transform for fluorescent images (Padfield, Rittscher et al. 2011). Some software packages (such as ImageJ) propose plugins performing image registration.

The size of the acquisition grid will dictate the spatial accuracy of the phenomenon observed. The larger the grid size, the bigger the area of the sample that is observed, and the more representative the data is. This is because there is an intrinsic variability within a cell population growing on a surface due to a random distribution of the cells when seeding at the beginning of a cell culture and also as a result of cell movements afterwards. The grid size can thus have an effect on cell properties measured by automated image processing, from cell counts to cell morphological features. For example, the effect of the acquisition grid on estimated cell counts from the analysis of microscopic images was assessed in 96 well plates imaged at 10X (Brinkmann, Lutkemeyer et al. 2002). The authors showed that the measure was reliable even if only 3 images per well were

analyzed (Brinkmann, Lutkemeyer et al. 2002), but that would not be necessarily the case at higher magnifications. A last consideration for setting-up the acquisition grid size is specific requirements for image processing. For instance, the acquisition grid size can be adjusted if a subsampling of the images is required in order to perform a quadtree decomposition to classify image patches into different categories for area segmentation (Qian, Peng et al. 2009).

The size of the acquisition grid will influence the temporal imaging frequency (the larger the grid size, the lower the imaging frequency achievable). The imaging frequency is defined as being the number of times the microscope can image the whole acquisition grid per unit of time in a live cell imaging experiment. To measure a specific phenomenon, the imaging frequency must be adapted in order to capture as much information as possible in the shortest time while providing as much spatial information as possible (scanning the largest area). So there is a tradeoff between the size of the acquisition grid (spatial information resolution) and the imaging frequency achievable (temporal resolution).

2.3.4.3 Illumination and exposure time

Illumination intensity is important in microscopy and can be adjusted on most microscopes by tuning the voltage of the light source. Illumination intensity should be high enough to be able to differentiate cellular from background regions in an image (higher contrast, (Cox 2007)). Two manipulated variables are available in order to ensure a proper contrast for image processing: light source voltage (illumination intensity) and camera exposure time. Some automated microscope offer the possibility to adjust automatically the light source voltage and the camera exposure time by an algorithm using an objective function. Since cells are very sensitive to light exposition (also NIR light can heat the culture medium), the intensity of the light source should be as low as possible. To compensate, it is needed to increase the exposure time of the camera, thus increasing the time required to acquire each image (this can have a detrimental effect on imaging frequency). As a consequence, there is a tradeoff between illumination intensity and exposure time. The exposure time required for a particular situation will be highly influenced by the performance of the camera CCD (related with the signal-to-noise ratio associated to the status of the CCD (Cox 2007)).

Illumination variations within a single image are a common challenge for microscopic image processing, and sometimes require preprocessing steps. A potential problem is with

the light coming from non-focal planes making its way to the CCD of the camera (mostly in fluorescence microscopy). To remove the impact of out-of-focus light in microscopic images, deconvolution techniques are required. For optical phase images (mostly PCM), the source of intensity variations may be caused by the shadowing of steep-sided culture wells, meniscus irregularities and non-uniformity of the plastic base used in culture plates (Curl, Harris et al. 2004). These sources of intensity variations are unavoidable in live cell imaging experiments. As a result contaminated images may have large brightness and contrast changes (Xiong, Chia et al. 2011). For phase contrast microscopy in particular, one of the main challenges is the “halo” effect, which causes spurious bright areas around phase objects that induce large phase shifts (Otaki 2000). When using PCM to image multiwell plates, the liquid meniscus (related with the surface tension of the culture medium) can have a strong effect on illumination variations (DeVaney 2008). In fact, the liquid meniscus acts as an optical lens, deviating incident light coming thru a well. Hence, as we move away from the center of the well, the images are getting brighter, and we loose the phase contrast (the images are similar to brightfield images). This problem is illustrated in a mosaic in Figure 2-15, where 1521 images were acquired to image an entire well of a 24 wells multiwell plate in PCM (20X): it is possible to see the loss of phase contrast from the center towards the periphery of the well.

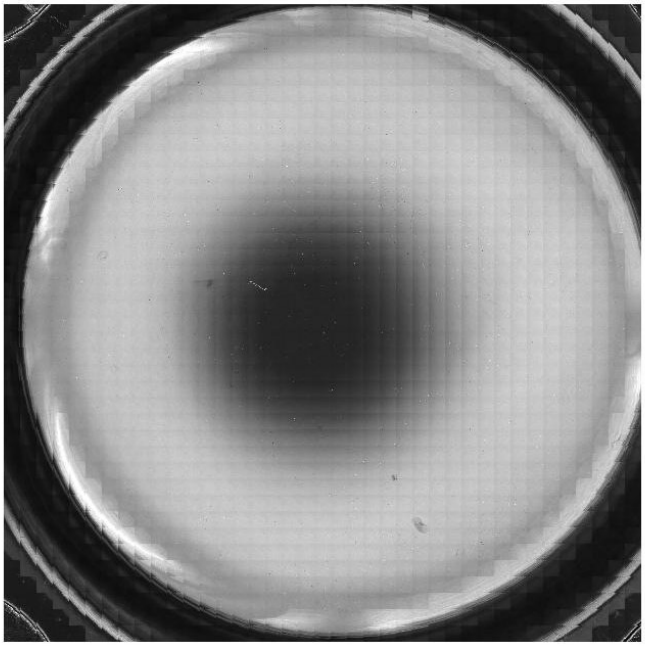


Figure 2-15 : Mosaic of 1521 images acquired to image an entire well of a 24 wells multiwell plate in PCM (20X): there is a loss of phase contrast at the well periphery.

Image processing techniques (background correction, deconvolution techniques) can be used to compensate for illumination problems. Another way is to use hardware modifications to get rid of those problems. For example, plate inserts are available on the market and can be put inside each well of a multiwell plate to get rid of the meniscus effect (DeVaney 2008). To reduce the artifact halo effect around cells, one can perform a hardware modification on the objective of the microscope via apodization (Otaki 2000). An apodized phase plate consisting of a ¼ wavelength phase shift ring with a transmittance of 25% is used with a pair of adjacent rings which have a transmittance of 50% (Otaki 2000). The apodized phase plate is placed at the back focal plane of the objective, reducing the halo of phase objects and providing enhanced inner details. Another approach is to use the automated features of the microscope to optimize the illumination set-up frame by frame. For example, it is possible to take a series of images with different exposure times and to use a high dynamic range approach to highlight the cells and the information in a same summary image (HRD image), even though there are huge illumination differences throughout the same well (Xiong, Chia et al. 2011). In this method, a summary image (HDR image) is created by combining multiple-exposure images for each field of view followed by a background subtraction using a Gaussian blurred image (to reduce the effect of uneven background illumination) and finally cells are segmented using a level set method (Xiong, Chia et al. 2011).

2.3.4.4 Focus adjustment

Focusing can be defined as adjusting the distance setting of a lens in order to produce a sharp image on a camera CCD (MIR 1999). In that sense, a well-focused image will ensure that the objects captured will have sharp contours as an out-of-focus image will have objects with blurry contours. Adjusting the objective height to have the right focus is important when imaging adherent cells using an inverted microscope. It is required to tune the focus properly to ensure that the details associated with the cells and their boundaries are not fuzzy, and that most of the light reaching the camera CCD is coming from the cells. Figure 2-16 shows examples of images taken at the right and wrong focus. Taking images at the wrong focus can have an impact on the accuracy of the results, since fuzzy cells might not be detected, cells boundaries can hardly be identified and the texture inside cells cannot be analyzed properly.

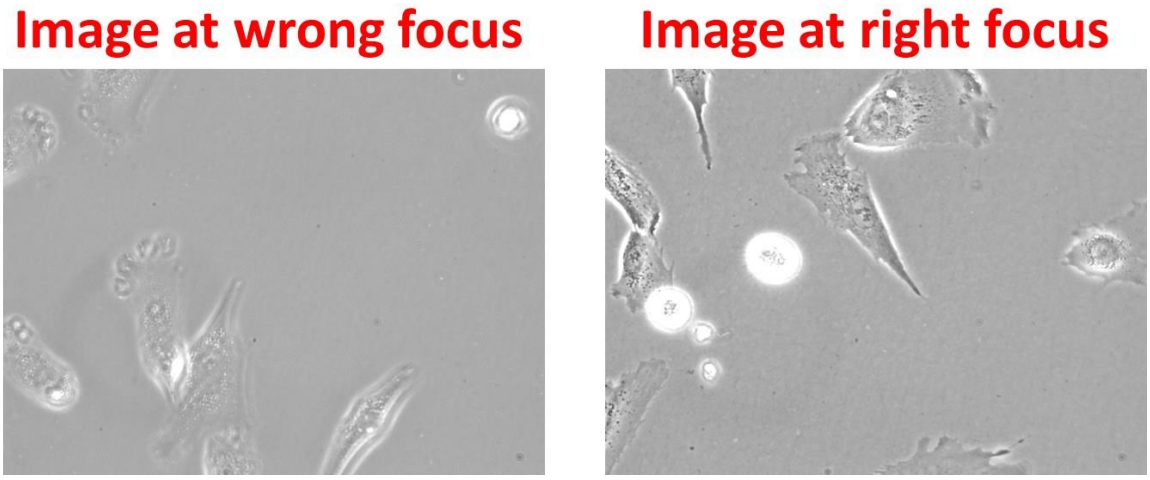


Figure 2-16 : Examples of images taken at the wrong and right focus (myoblasts imaged at 40X in PCM)

When using a large acquisition grid, the focus plane can change between fields of view because of different factors. A non-uniform focus profile between fields of view can be caused by a tilt in the microscope multiwell plate or slide, by a non-uniform thickness of the sample holder or the sample itself and by the presence of a meniscus in the liquid. For example, an optimal focus profile at the bottom of a well in a 24 wells multiwell plate obtained by an autofocus algorithm (Metamorph®) is shown in Figure 2-17.

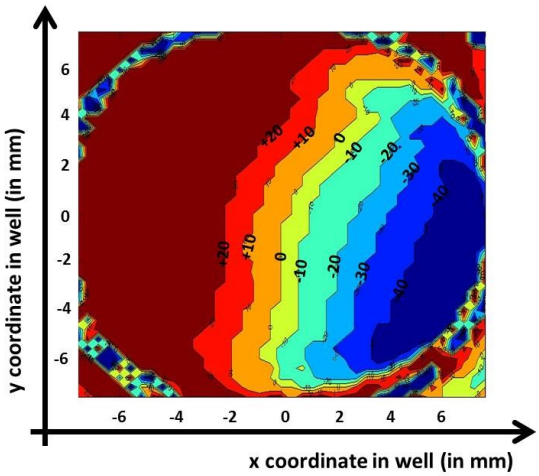


Figure 2-17 : Example of an optimal focus profile at the bottom of a well in a 24 wells multiwell plate. At each well location (for x and y coordinates), the Metamorph® autofocus utility found the optimal focus. The optimal focuses are provided as a topographic map, with isolines providing the z-coordinate level (focus plane height).

2.3.4.4.1 Autofocus algorithms

From Figure 2-16, it is possible to see that usually out-of-focus images are associated with objects that lose their texture (the cells are fuzzy, and we do not see the fine details inside them), and this can be used to detect images that are not at the right focus. For example, Padfield used wavelets to detect out-of-focus images by looking at the loss of texture and by computing a defocus ratio (Padfield, Rittscher et al. 2011). Since focus has an impact on image texture analysis, using an autofocus algorithm to ensure that each image is acquired at the right focus is really important to ensure data quality. In this work, the autofocus algorithm readily available in Metamorph® was used (unfortunately no details were provided by the manufacturer about the algorithm used by the software). Most autofocus algorithms can be classified into four categories: derivative-based algorithms (assuming that well-focused images have more high-frequency content, including gradient methods and wavelet methods), statistics-based algorithms (using variance and correlation), histogram-based algorithms (using the histogram of an image) and intuitive algorithms (mainly based on thresholding) (Yu, Duthaler et al. 2005). An autofocus algorithm generally calculates some features in the image as the microscope depth is shifted (adjusting the height of the objective for inverted microscopes), and the global maximum (or minimum) of a curve fitted onto the calculated feature values is chosen as the true focal depth or the right focus (Dashan, Padfield et al. 2010). With an autofocus application, there is a risk that some tiles in a whole acquisition grid might not be at the best focus because of local variations. A method was suggested to detect out-of-focus tiles by looking at the variation of features calculated in the overlapping regions between adjacent tiles (Dashan, Padfield et al. 2010). Wavelets are particularly suitable to assess focus quality, since the blurring in out-of-focus images increases energy in low-pass bands and decreases energy in high-pass bands (Hui, Weibin et al. 2006).

Focus adjustment in hyperspectral microscopy can be very difficult, certainly for systems using a “push broom” hyperspectral camera with one spatial and one spectral dimension. This is due to the fact that one needs to move the sample to obtain an image. It is hard to assess focus quality without looking at overall image quality and textural details. A solution is to put the system in “autofocus mode” and to acquire several hyperspectral cubes at different focal planes one after the other until the right focus is identified by analyzing each image obtained (Katari, Wallack et al. 2009). Another method is to manually change de focus by looking at one specific spatial line (hopefully capturing fine details) in a

hyperspectral image (slice of the hyperspectral cube) and to choose the focus that gives the most precise details (“sharp” details or refined texture) (Polerecky, Bissett et al. 2009).

2.4 Image processing

2.4.1 Definition of a digital image

A grey level digital image is a rectangular array of measured values of some property of interest collected from the area under study, and in which the array elements are called pixels (short for picture elements) (Rodenacker and Bengtsson 2003). Each pixel represents the integrated intensity of light measured over the spatial extent of the pixel, which depends on the optical and spectral properties of the imaging system. The measured intensity is proportional to the quantity of light transmitted/reflected/diffracted (with or without interference) or emitted by fluorescence. The image array has at least two spatial dimensions, but since all objects have intrinsically three dimensions, it is common to capture 3D images via confocal, deconvolution or multiphoton fluorescence microscopy (Rodenacker and Bengtsson 2003). Figure 2-18 shows an example of a typical 2D digital image.

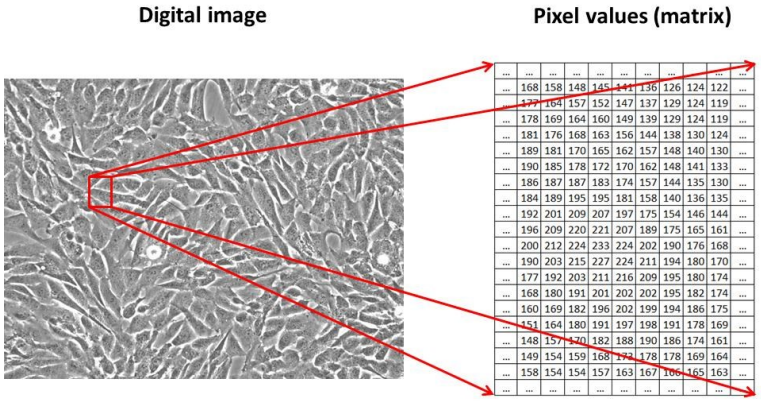


Figure 2-18 : Example of a 2D digital image (myoblasts imaged in PCM at 20X)

An additional dimension can be obtained by limiting the spectral range over which a sensor registers light, providing the capacity to obtain more or less well resolved spectra of a two dimensions image (spectral image) (Rodenacker and Bengtsson 2003). The most common spectral image is the RGB image, having three spectral channels: red, green and blue. In fluorescence microscopy, it is possible to use narrow bandwidth limiting filters that can match the property of specific stains, and the superposition of the different images acquired with different filters can provide a spectral image with a certain number of

spectral channels (Rodenacker and Bengtsson 2003). Alternatively, the “push broom” set-up can be used to acquire spectral images (hyperspectral cubes) of transmitted or emitted light with varying spectral resolutions. Figure 2-19 shows an example of a hyperspectral image having two spatial dimensions and one spectral dimension. For example, a spectral cube of myoblasts grown in SFM (imaged in PCM at 20X) was acquired using a push broom set up (slices in the y and λ dimensions), and the information is summarized in a T_1 score image generated by Principal Component Analysis (PCA).

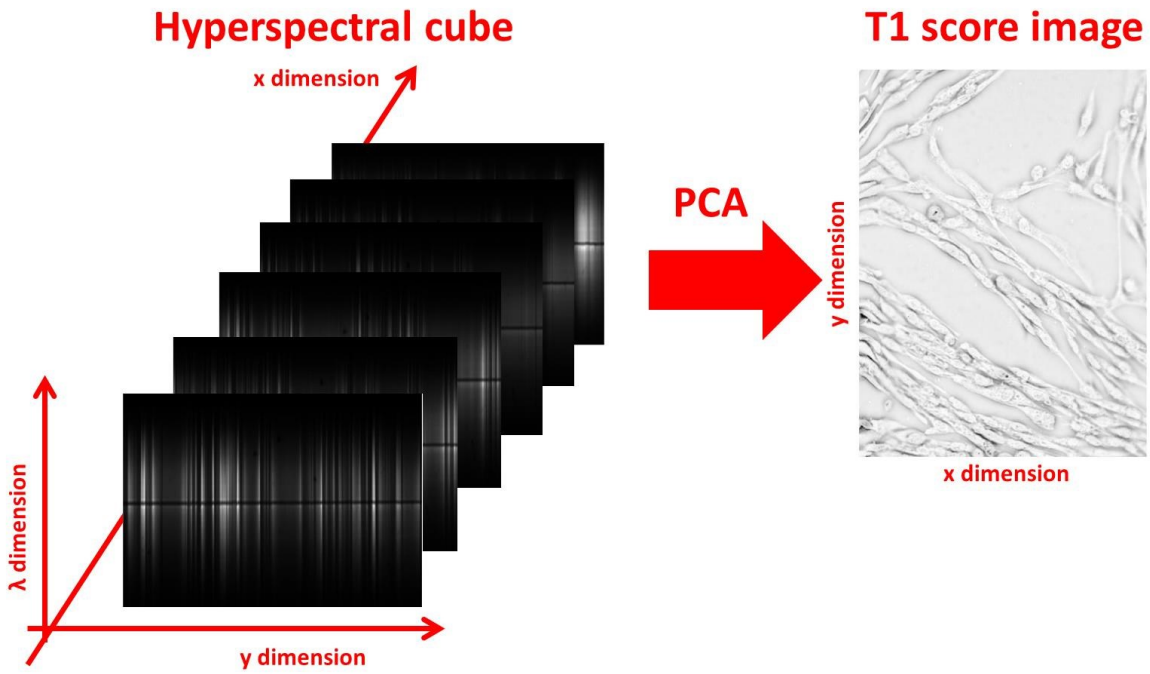


Figure 2-19 : Example of a hyperspectral image, with two spatial dimensions and one spectral dimension. The information is summarized via a T_1 score image generated by Principal Component Analysis (PCA).

Another dimension that can be considered is time. If the specimen is subjected to processes changing it through time, it is possible to acquire successive images spaced at proper time intervals to monitor these changes. When performing live cell imaging, digital images can be processed statically or dynamically (considering the images acquired as a time series) (Miura 2005).

A digital image can then possess up to 5 dimensions (three spatial, one spectral and one temporal dimension) (Rodenacker and Bengtsson 2003). For each image, one can

analyze these different dimensions to extract features and important information via image processing techniques.

2.4.2 Fourier transform of a digital image

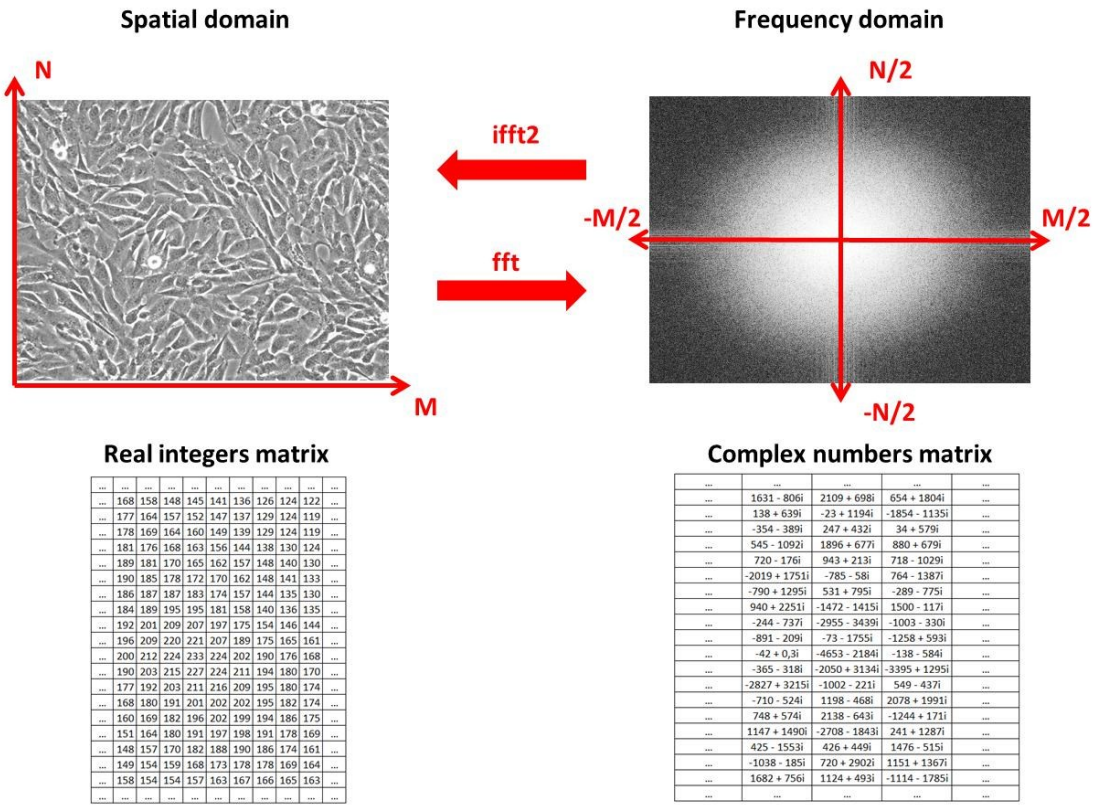
The Fourier transform of an image consists of a transformation from a spatial domain to a frequency domain. Since no information is lost or added (simply a change of referential), this transform is reversible. For a digital image f(x,y) of size M by N in the spatial domain, its discrete Fourier transform (DFT) can be computed by the following equation (Gonzalez and Woods 2008):

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2-12}$$

In which u and v are respectively horizontal and vertical spatial frequencies (in cycles) and x and y are respectively horizontal and vertical spatial coordinates (in pixels). Since the Fourier transform is reversible, the inverse discrete Fourier transform (IDFT) from F(u,v) can be calculated by the following equation (Gonzalez and Woods 2008).

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2-13}$$

To compute this transform on 2D images, the functions fft2 and ifft2 are used in Matlab®. An example is provided in Figure 2-20 for an MxN image.



...	...	...	...	...	...	...	...	...	...	...
...	168	158	148	145	141	136	126	124	122	...
...	177	164	157	152	147	137	129	124	119	...
...	178	169	164	160	149	139	129	124	119	...
...	181	176	168	163	156	144	138	130	124	...
...	189	181	170	165	162	157	148	140	130	...
...	190	185	178	172	170	162	148	141	133	...
...	186	187	187	183	174	157	144	135	130	...
...	184	189	195	195	181	158	140	136	135	...
...	192	201	209	207	197	175	154	146	144	...
...	196	209	220	221	207	189	175	165	161	...
...	200	212	224	233	224	202	190	176	168	...
...	190	203	215	227	224	211	194	180	170	...
...	177	192	203	211	216	209	195	180	174	...
...	168	180	191	201	202	202	195	182	174	...
...	160	169	182	196	202	199	194	186	175	...
...	151	164	180	191	197	198	191	178	169	...
...	148	157	170	182	188	190	186	174	161	...
...	149	154	159	168	173	178	178	169	164	...
...	158	154	154	157	163	167	166	165	163	...
...	...	...	...	...	...	...	...	...	...	...

...	...	...	...	...
...	1631 - 806i	2109 + 698i	654 + 1804i	...
...	138 + 639i	-23 + 1194i	-1854 - 1135i	...
...	-354 - 389i	247 + 432i	34 + 579i	...
...	545 - 1092i	1896 + 677i	880 + 679i	...
...	720 - 176i	943 + 213i	718 - 1029i	...
...	-2019 + 1751i	-785 - 58i	764 - 1387i	...
...	-790 + 1295i	531 + 795i	-289 - 775i	...
...	940 + 2251i	-1472 - 1415i	1500 - 117i	...
...	-244 - 737i	-2955 - 3439i	-1003 - 330i	...
...	-891 - 209i	-73 - 1755i	-1258 + 593i	...
...	-42 + 0,3i	-4653 - 2184i	-138 - 584i	...
...	-365 - 318i	-2050 + 3134i	-3395 + 1295i	...
...	-2827 + 3215i	-1002 - 221i	549 - 437i	...
...	-710 - 524i	1198 - 468i	2078 + 1991i	...
...	748 + 574i	2138 - 643i	-1244 + 171i	...
...	1147 + 1490i	-2708 - 1843i	241 + 1287i	...
...	425 - 1553i	426 + 449i	1476 - 515i	...
...	-1038 - 185i	720 + 2902i	1151 + 1367i	...
...	1682 + 756i	1124 + 493i	-1114 - 1785i	...
...	...	...	...	...

Figure 2-20 : Example of a Fourier transform of a 2D digital image of dimension M by N

It is important to note that the Fourier transform of a digital image generates a matrix of complex numbers. The maximum frequencies associated with a Fourier transform of an image are M/2 and N/2 cycles, since higher frequencies are associated to aliasing (process in which high frequencies in a signal are assimilated to lower frequencies due to under-sampling according to the Nyquist criterion (Gonzalez and Woods 2008)). The maximum frequency in an image corresponds to a series of black and white pixels alternating every two pixels. Note that the frequencies u and v in the frequency domain may have various units (cycles for the entire image length or height, cycles per pixel and rad per pixel) as needed. For instance, the maximum horizontal frequency is M/2 cycles for the entire image length, corresponding to 0.5 cycles per pixel or π rads per pixel.

Noteworthy, since M and N are different (most images are not square in size), the frequency resolution (difference in frequency between two adjacent points in the FFT transform of an image) is not the same for u and v (Gonzalez and Woods 2008). For

instance, the horizontal frequency resolution (Δu) is 1/M cycles per pixel and the vertical frequency resolution (Δv) is 1/N cycles per pixel.

The Fourier transform matrix can be processed to compute the magnitude (Mag), the phase angle (PA) and the power spectrum (PS) associated with the complex numbers (Gonzalez and Woods 2008). The formulas are given as follows (Gonzalez and Woods 2008), for which Re(u,v) and Im(u,v) refer to the real and imaginary portions of the Fourier transform matrix F(u,v):

$$\mathrm{Mag}(u,v) = \sqrt{\mathrm{Re}(u,v)^2 + \mathrm{Im}(u,v)^2} \tag{2-14}$$

$$\mathrm{PA}(u,v) = \arctan\left[\frac{\mathrm{Im}(u,v)}{\mathrm{Re}(u,v)}\right] \tag{2-15}$$

$$\mathrm{PS}(u,v) = \mathrm{Re}(u,v)^2 + \mathrm{Im}(u,v)^2 \tag{2-16}$$

Figure 2-21 shows the Mag (Magnitude), PS (Power spectrum) and PA (Phase angle) matrices for a typical live cell image acquired in phase contrast microscopy (PCM).

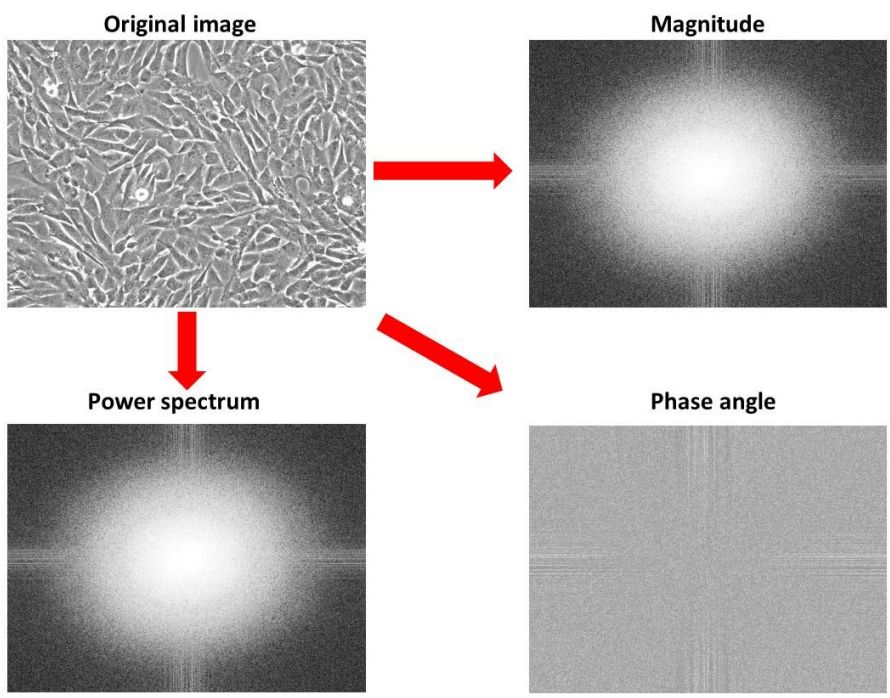


Figure 2-21 : Magnitude, power spectrum and phase angle matrix associated to the DFT of a typical PCM image

The power spectrum (similar to the magnitude) and the phase angle provide information about the signal frequency profile present in an image, and the orientation of the signal. Figure 2-22 and Figure 2-23 show the phase angle and power spectra matrices of typical images of myoblasts grown in SSM and SFM (both media described in Table C-1 and in Table C-2). It is clearly shown that cells grown in SFM are associated with signals of higher frequencies (higher values that are located further away from the power spectrum center). The power spectrum can also detect the orientation of the signal in each image (higher values perpendicular to the orientation of the cells at high degrees of confluence).

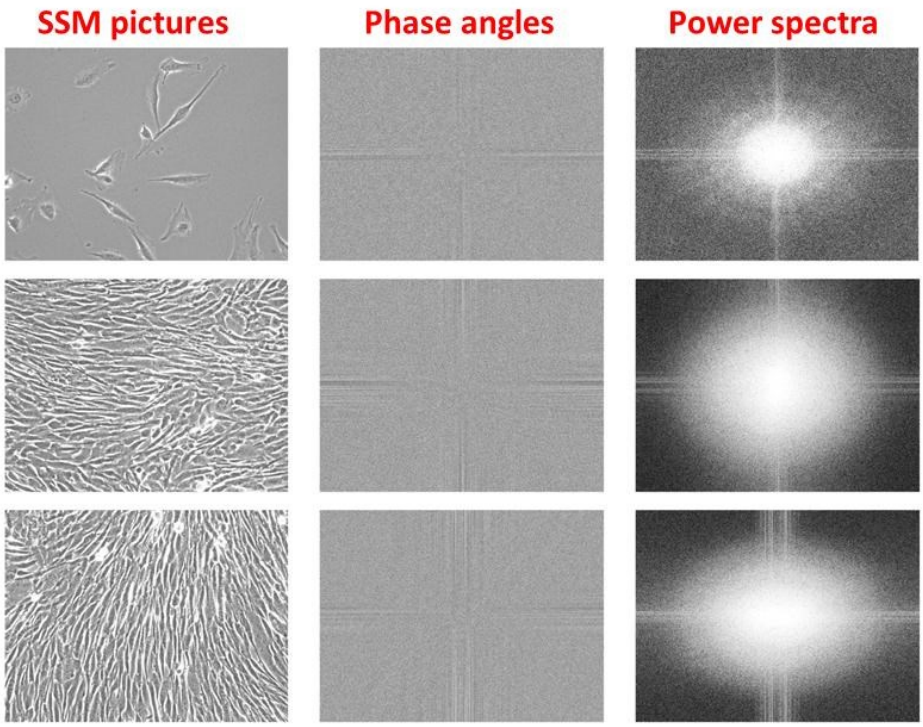


Figure 2-22 : Phase angle and power spectra matrices for images of cells grown in SSM

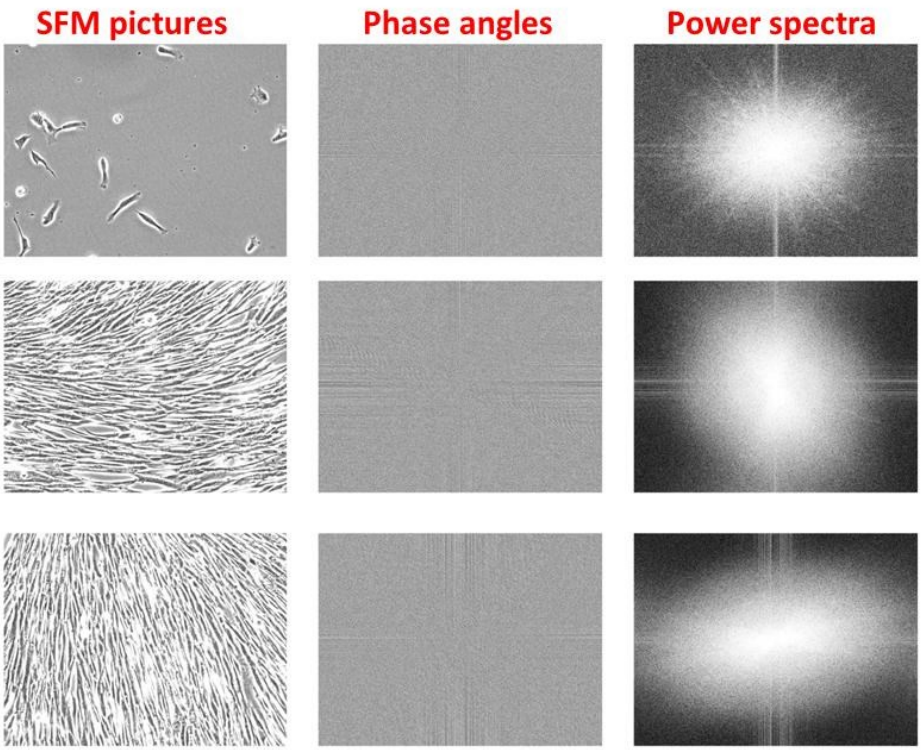


Figure 2-23 : Phase angle and power spectra matrices for images of cells grown in SFM

2.4.3 Image pre-processing

Since microscopic images can be contaminated by uneven illumination or by out-of-focus light, image pre-processing can be used to enhance their quality before further processing. PCM microscopy is a technique that converts small phase shifts caused by transparent specimens into light amplitude variations that can be captured by the observer or by the CCD of the camera (An-An, Kang et al. 2012). Due to the optical principle and the imperfections due to the phase shifting process, PCM provides images containing several artifacts such as halos (local contrast reversal surrounding the specimen) and shade-offs (gradual bending of the intensity profile of a large flat specimen from the edges to the center) (An-An, Kang et al. 2012). Removing these artifacts from microscopic images can facilitate automated image processing in order to perform cell segmentation and cell tracking (Zhaozheng and Kanade 2011). Pre-processing techniques available can be roughly classified as background correction, spatial filtering and frequency filtering.

2.4.3.1 Background correction

When processing a microscopic image, there is an incentive to remove information related with the background (spatial intensity gradients due to uneven illumination, optical imperfections of culture plates, small cellular debris, etc.) in order to identify group of pixels belonging to cells. Several background correction strategies are reported in the literature.

2.4.3.1.1 Brightfield and phase contrast microscopy

The main problem in brightfield and in phase contrast microscopy images is uneven illumination, which creates brighter and darker regions in microscopic images (spatial intensity gradients). Background correction can be used to compensate by dividing each image by a smooth filtered version of itself (Kang, Miller et al. 2006). Another method is to acquire a reference image (an "empty" image taken under the same conditions) and to subtract it from the original image (Burgemeister, Nattkemper et al. 2010). To amplify pixels associated to high intensity differences (most likely associated to cells) and to dim out pixels with low brightness associated to the background, one can also use a sigmoidal function (Burgemeister, Nattkemper et al. 2010).

To correct images for uneven illumination, morphological operations such as Top-hat and Bottom-hat filters can be used (Gonzalez and Woods 2008; Kachouie, Fieguth et al. 2010). To subtract the background of PCM images, a filter called the rolling ball algorithm (also called Top-hat filter) is commonly used (Polzer, Haasters et al. 2010). In the rolling ball algorithm, a local background value is determined for every pixel by averaging over a very large ball neighborhood (structuring element) around the center pixel. This value is hereafter subtracted from the original image for each pixel, removing low frequency spatial variations associated with the background. The radius of the ball should be set to at least the size of the largest object in the image (Sternberg 1983).

As a background correction strategy, the histogram of the image can be modified. For example, as a pre-processing step for analyzing PCM images, it is possible to stretch the contrast and to perform histogram equalization (Sakuma and Tanaka 2011). If a series of images is acquired through live cell imaging experiments, both the images histogram and temporal information can be used to spot background pixels, since their intensity will remain fairly constant through time compared to pixels associated to cells (Kachouie, Fieguth et al. 2007).

2.4.3.1.2 Hyperspectral microscopy

Background correction/removal is an inherent difficulty in hyperspectral microscopy. Several techniques are proposed in the literature. For instance, it is possible to use two calibration images, one dark where no light is reaching the CCD of the camera (noise or dark current) and one white reference image for which the specimen support is used, but without the cells (Siddiqi, Li et al. 2008). This method is used to remove the effect of uneven illumination, the effect of scanning lines, and the impact of the lamp, the culture medium and the glass/plastic support on the cellular reflectance or transmittance spectra measured (Siddiqi, Li et al. 2008). For reflectance images, it was also proposed to compute the ratio of the specimen images to a white reference image using a 99% reflectance standard such as Spectralon (Polerecky, Bissett et al. 2009). For transmittance images acquired in hyperspectral microscopy, it is recommended to compute the ratio of the specimen images to a white reference image (empty specimen support) (Polerecky, Bissett et al. 2009).

To correct the spectra measured by a hyperspectral camera in a confocal microscope, it is possible to perform a PCA on a dark image (dark hyperspectral cube) and to subtract the rank-one PCA dark image obtained from each measured hyperspectral cube (Haaland, Jones et al. 2009). For a more uniform analysis, it is recommended to correct the hyperspectral cubes for a noise following a Poisson distribution and a CCD submitted to a Gaussian noise (Haaland, Jones et al. 2009). To remove noise, a threshold can be used to set to zero any pixel value associated to a signal-to-noise ratio lower than a specified limit (Cutler, Haaland et al. 2009).

2.4.3.2 Spatial filtering

Spatial filtering (commonly called “convolution” in the literature) is a very versatile operation that has a broad spectrum of applications in image processing (Gonzalez and Woods 2008). A spatial filter consists of a neighborhood (usually a small rectangle) and a predefined operation that is performed on the image pixels encompassed by the neighborhood (Gonzalez and Woods 2008). The filtering operation creates a new value in the filtered matrix with the coordinates corresponding to the center of the neighborhood whose value is a function of the values in the neighborhood (Gonzalez and Woods 2008). The filtered image is generated as the center of the neighborhood (filter window) visits each pixel in the image. Figure 2-24 shows a raw image filtered with an average filter of size N1 by N2 (neighborhood).

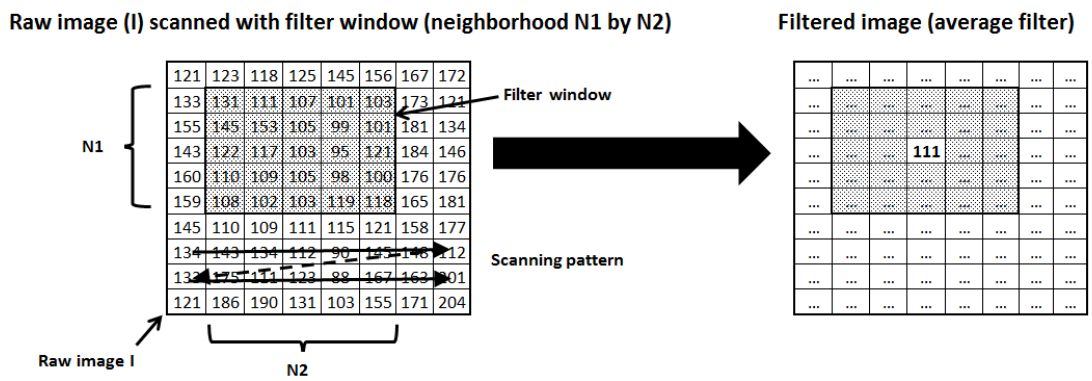


Figure 2-24 : Spatial filtering using the average filter of size N1 by N2

If the operation performed on the raw image pixels is linear, then the filter is linear (Gonzalez and Woods 2008). A typical linear filter of size m by n applied on a raw image can be expressed by the following image (Gonzalez and Woods 2008):

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \tag{2-17}$$

Where g(x,y) is the filtered matrix obtained from the raw image f(x,y), and w(s,t) are the filter coefficients (scalars) associated to the window of size m = 2a+1 by n = 2b+1.

On the other hand, if the operation on the pixel values captured by the filter is not linear, then it is called a non-linear filter. Order-statistic filters (median, entropy, range and standard deviation filters) are a common class of non-linear filters, since they are based on the ranking of the pixels encompassed by the filter window (Gonzalez and Woods 2008).

2.4.3.3 Frequency filtering

For linear filters, there is a direct relationship between filtering in the spatial domain and filtering in the frequency domain provided by the convolution theorem (Gonzalez and Woods 2008). For an image f(x,y) associated with its Fourier transform F(u,v), and a linear filter h(x,y) associated to its own Fourier transform H(u,v), the convolution theorem is expressed as follows (Gonzalez and Woods 2008):

$$f(x,y) \vartriangle h(x,y) = F(u,v).* H(u,v) \tag{2-18}$$

$$f(x,y).* h(x,y) = F(u,v) \vartriangle H(u,v) \tag{2-19}$$

In which $\triangle$ is the convolution operator for two matrices and ".*" is the element-by-element multiplication of two matrices. This means that the convolution of an image in the spatial domain with a linear filter is equivalent to the element-by-element multiplication of the Fourier transforms of the image and that of the linear filter. It is then possible to obtain the filtered Fourier transform of the image G(u,v) by the following equation:

$$G(u,v) = F(u,v).*H(u,v) \tag{2-20}$$

Since this is an element-by element multiplication, convolving an image with a linear filter can be more efficient in the frequency domain than in the spatial domain, for which it is necessary to scan spatially the whole image, pixel by pixel, and to compute a function from each pixel's neighborhood. Even though it is necessary to compute the Fourier and the inverse Fourier transform of the product, the Fast Fourier Transform (FFT) algorithm is already optimized to minimize computational costs and to be very efficient (in comparison to performing the convolution in the spatial domain). Figure 2-25 shows the example of a cell image convolved with a Gaussian low pass filter (linear) in the frequency domain in order to remove high frequency noise.

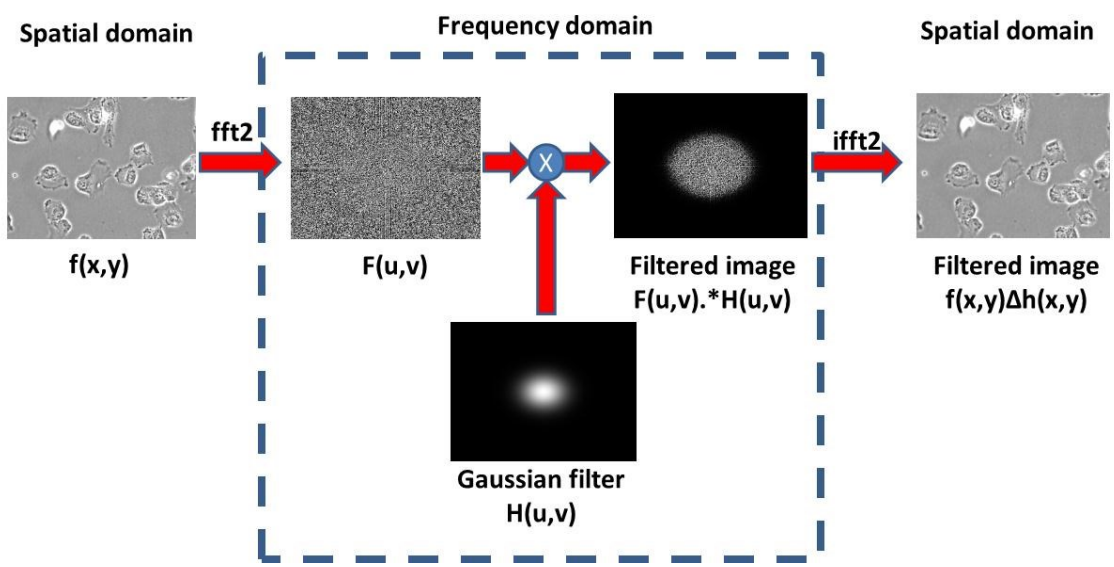


Figure 2-25 : Convolution of an image with a linear (Gaussian) filter in the frequency domain

In an optical imaging system (when using a microscope for instance), if the true image is contaminated (convolved) with a linear degradation function that may be related with the system itself or to unwanted signal from out-of-focus light (named point spread function or

"PSF" (Cox 2007)), the Fourier transform of the real image F(u,v) is contaminated by a degradation function D(u,v) and the system acquires a contaminated image C(u,v):

$$C(u,v) = F(u,v).* D(u,v) + N(u,v) \quad (2\text{-}21)$$

Where N(u,v) is an extra term to take into account interference noise (Gonzalez and Woods 2008). Since image degradation can be modeled as a result of convolution, image restoration techniques usually try to find a proper linear filter (good estimate of D(u,v)) that can invert this process, thus recovering the real image F(u,v) from the contaminated image C(u,v). This method is called "image deconvolution" (Gonzalez and Woods 2008).

Since noise is usually associated with high frequencies in an image and background with low frequencies, it is possible to filter out details in an image using low pass, high pass or band pass filters (depending on the situation). For example, to filter background noise in fluorescent images, the undecimated wavelet transform (with a hard threshold on the detail coefficients) can be used (Olivo-Marin 2002). Alternatively, one can use the Fourier transform as a tool to differentiate noise from signal patterns in spectral images (Sanchez, Prow et al. 2010).

2.4.3.3.1 Brightfield and phase contrast microscopy

By looking at the phase contrast optics (mathematical modelling), PCM images can be relatively well explained by a linear imaging model, and it is possible to use a quadratic function to restore the authentic phase contrast images without artifacts such as halos and shade-offs (Yin, Li et al. 2010). The cells in the restored images ($\hat{F}(u,v)$) could be segmented easily by simple thresholding (Yin, Li et al. 2010). It is possible to restore a PCM or a DIC image by minimizing a regularized quadratic cost function that is adaptable to input image properties, considering a whole image sequence (Zhaozheng and Kanade 2011). In order to do so, an obscured airy kernel is used for phase contrast microscopy and a Difference-of-Gaussian kernel is used for DIC images in an iterative deconvolution algorithm (Zhaozheng and Kanade 2011). Under a positive phase contrast microscope, adherent stem cells grown in culture appear as dark objects surrounded by bright halos, and the phase contrast image can be approximated by a linear imaging model (An-An, Kang et al. 2012). In phase contrast microscopy, image formation can be seen as a linear process with the following degradation function (Yin, Li et al. 2010):

$$D(u,v)_{\text{PHASE CONTRAST}} = \delta(u,v) - \text{airy}\left(\sqrt{u^2 + v^2}\right) \quad (2\text{-}22)$$

This approach could be interesting for cases in which the optical path stays relatively the same. However, as in the case of this work, since several locations in several wells are imaged, this degradation function is much likely not constant (as it is the case for the optimal focus). This is due to the non-uniformity of the cell culture plate and the meniscus effect.

2.4.3.3.2 Hyperspectral microscopy

When analyzing hyperspectral cubes obtained in fluorescence microscopy, one of the major problems is related with the noise generated by fluorescent molecules that are out-of-focus. In an epifluorescence microscope, thick specimens can produce an image that represents the sum of the in-focus details convolved with the blurry out-of-focus details (Monici 2005). This effect is not too significant for low magnification images (10X or lower), for which the depth-of-field is important (Monici 2005). For higher magnifications, several methods were developed to get rid of the noise in thicker specimen, such as confocal microscopy, multiphoton microscopy and image deconvolution for epifluorescence microscopy. These methods provide the opportunity to perform an optical sectioning of the noise generated by out-of-focus light, avoiding a physical sectioning of the specimen during its preparation (making the specimen thinner) (Monici 2005). In a fluorescence image, the intensity of each pixel is linearly related by convolution to the fluorescence intensity of each point in the specimen (Monici 2005). As a consequence, the influence of each element in the optical path (filters, lenses, etc.) can be removed by a deconvolution algorithm that can compensate for the loss of contrast and for the blurring caused by noise (Monici 2005). The advantage of using an epifluorescence microscope coupled to a deconvolution technique instead of using a confocal microscope (laser excitation light) is that the cells and tissues are exposed to a milder excitation light, thus making this technique less destructive (Monici 2005). When using fluorophores, deconvolution techniques can be very useful to deconvolve the autofluorescence signal of the cells from the main fluorescence signal (Constantinou, Dacosta et al. 2009).

2.4.4 Segmentation algorithms

Many segmentation algorithms are described in the literature for segmenting/highlighting cells in microscopic images. By definition, a segmentation algorithm subdivides an image into its constituent regions or objects (usually cells for microscopic images), and the extent to which the subdivision is carried out depends on the problem to be solved (Gonzalez and

Woods 2008). Segmenting the image into relevant objects and background regions is a critical and difficult step in each image processing scheme, as the objects or regions need to be analyzed further to make important conclusions (Rodenacker and Bengtsson 2003). For cytological applications, a whole range of segmentation algorithms is available, from simple thresholding techniques to the extraction of features for each pixel followed by their classification into different classes or sub-regions (Rodenacker and Bengtsson 2003). The choice of the features depends on the characteristics of the sub-areas to be discriminated (Rodenacker and Bengtsson 2003).

As a result of the segmentation process, each object identified is linked to a mask *O*, which is the set of all pixels *p=(x,y)* that are included in the mask ($p \varepsilon O$) (Rodenacker and Bengtsson 2003). It is also possible to define $C \subset O$ as the ordered set of pixels of the contour or border of the object (Rodenacker and Bengtsson 2003). As a result, after a segmentation process, each set *O* describes the zone of measurement and *C* describes the border of the object as a list of coordinates (Rodenacker and Bengtsson 2003). From a computation perspective, the result of the segmentation process generates a Segmentation Map (SM), in which each pixel is assigned a class, as illustrated in Figure 2-26. In the Segmentation Map, two objects are identified (O_1 in red and O_2 in green), and their respective borders C_1 and C_2 are highlighted in blue. Background pixels are assigned the class 0 and are colored in black.

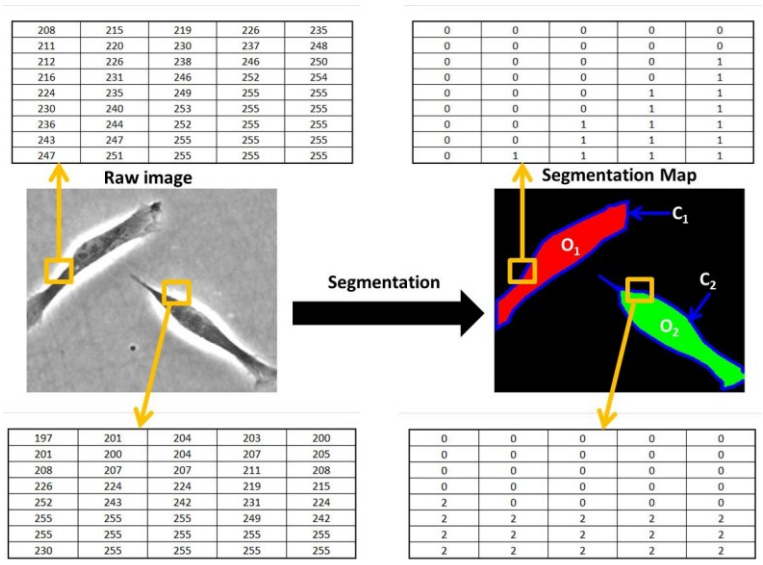


Figure 2-26 : Image segmentation process, in which a Segmentation Map (SM) is generated from a raw image

To perform object segmentation in images, Gonzalez et Woods (2008) classify roughly segmentation algorithms as they are based on edge detection, on region detection or on pattern recognition. An overview of these methods is presented in the following sections.

2.4.4.1 Edge detection methods

Edge detection methods are based on gradients between the contour of the objects to be segmented and the background. Many methods are using filters to detect edges, such as the Sobel filter, the Canny filter, the Prewitt filter and the Marr-Hildreth edge detector (Gonzalez and Woods 2008). The Canny filter algorithm can be quite effective since it uses two thresholds to detect strong and weak edges, and weak edges are only accepted if they touch a strong edge (Canny 1986). Gradient filters, which compute the local derivative of pixel intensities, can also be used to process PCM images, since cells and membranes are surrounded by a bright halo (strong gradients between the interior of the cells, the cells membrane and the background). Numerous applications of edge detection filters in live cell imaging are mentioned in the literature. For example, the contour of artificial phospholipid vesicules was segmented by looking at the maximum gradient in PCM images (Usenik, Vrtovec et al. 2011). A robust edge detection method using a local adaptive Gaussian kernel convolution was used to detect neural stem cells (Taoyi, Yong et al. 2010). To select the resolution at which a cell edge was the most pronounced, a multi-scale approach was used by downsampling by a factor of 2_k the original image followed by the computation of the intensity gradient matrix and finally by computing the magnitude of the upsampled gradient matrix having the same size than the original image (Theriault, Walker et al. 2012).

The most advanced techniques to detect object boundaries are active contours (Kass, Witkin et al. 1988) and level set methods (Sethian and Adalsteinsson 1997), in which an energy function related to the area around a closed curve that partitions an image is minimized iteratively, thus making the curve evolve until it converges and captures objects within the image (Chan and Vese 2001; Lankton 2009). The basic idea of both active contour models (or snakes) and level set methods is to evolve a curve by calculating the energy around it, subject to constraints within an image, in order to detect the objects boundaries (cells for example) (Zhou 2007). The main difference between active contours and level set methods is in their implementation and how the curve is computed. Both methods require an initial segmentation (boundary) at the first iteration. Also, active contours usually keep the same number of objects from one iteration to the other,

compared to level sets where contours can merge or be created (the number of objects detected in the image is not constant).

2.4.4.1.1 Level sets

In active contours (snakes), a parameterization of a closed curve discretized into *P* points is required. The curve is deformed iteratively from iteration *t* to iteration *t+1* in order to minimize an energy function (Zimmer, Labruyere et al. 2002; Miura 2005). However traditional edge-based active contours (in which the image energy is computed by the integral of a locally computed edge map along the contour) usually provide poor results on fluorescence microscopy images when the boundaries are fuzzy and sensitive to initialization (Cheng and Rajapakse 2009). Even if new active contours models have been developed in the last few years, level set methods (geometric active contours) are becoming more popular since they don't require an explicit parameterization and do not suffer from any constraints on the topology as snakes do (Cheng and Rajapakse 2009). The region-based approach used in many level set methods provides strong robustness to noise and allows segmentation of cells with blurred edges (Cheng and Rajapakse 2009).

A level set is a representation of a curve that doesn't require parameterization, and uses a mapping function $\phi(x,y)$ having the same dimensions than the original image ($I(x,y)$). Regions inside a contour have negative values, regions outside a contour have positive values and pixels on the contour have a value of zero (signed distance function (Lankton 2009)). Figure 2-27 shows the evolution of a contour through a level set representation applied to an inverted phase contrast microscopy image containing cells. The Matlab® algorithm used was available freely on the internet from Lankton (2009). A significant number of iterations was required to detect the contour of one cell (18,000), thus making this particular approach difficult to implement and to use for analyzing in real time PCM images (for cell culture control purposes) or even for HTS applications producing an important quantity of images.

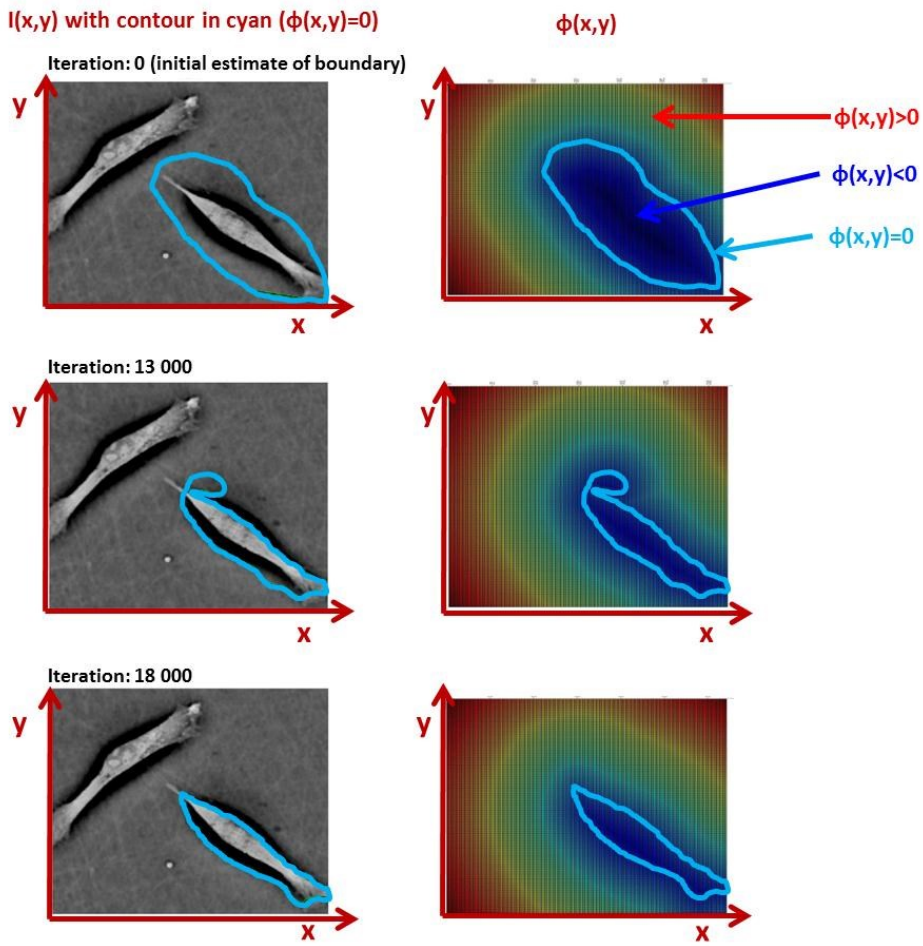


Figure 2-27 : Example of the evolution of a level set representation for segmenting a cell. A significant number of iterations is required to achieve a segmentation.

The goal is again to minimize the energy captured by the contour (E(ϕ(x,y))). The evolution of the level set will be influenced by the internal and external energy captured by the contour that will act as a force on the contour (F(ϕ(x,y))), as the energy is minimized. The evolution of the level set is based on its derivative, which is a function of different energies inside and outside the contour expressed as a force (Demirkaya, Asyali et al. 2009):

$$\frac{\partial \phi(x,y)}{\partial t} = F\big(\phi(x,y)\big)|\nabla\phi(x,y)| \qquad (2\text{-}23)$$

The level-set evolves from one iteration to the other (from iteration *t* to iteration *t+1*) with the following equation:

$$\phi(x,y)_{t+1} = \phi(x,y)_t + \left[\frac{\partial\phi(x,y)}{\partial t}\right]_t dt \tag{2-24}$$

The time/iteration interval *dt* must be adjusted to ensure that the level set evolves as fast as possible (minimize computation times), but not too fast to ensure convergence. Several energy-minimization functions (F(ϕ(x,y))) are reported in the literature. The simplest is based on the motion by mean curvature introduced by Osher and Sethian (Chan and Vese 2001):

$$\frac{\partial\phi(x,y)}{\partial t} = |\nabla\phi(x,y)|\mathrm{div}\left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right) \tag{2-25}$$

More complex approaches are also reported in the literature, depending on the energy functions and energy-minimization strategies used. For example, it is possible to use the image gradient to stop the evolution of the curve (Chan and Vese 2001):

$$\frac{\partial\phi(x,y)}{\partial t} = |\nabla\phi(x,y)|\left[-\nu + \frac{\nu}{\max(G(x,y))-\min(G(x,y))}(|G(x,y)| - \min(G(x,y)))\right] \tag{2-26}$$

Where *v* is a constant (tuning parameter), and G(x,y) is defined as the gradient of the convolution between a Gaussian filter g(x,y,σ) and the raw image I(x,y):

$$G(x,y) = \nabla[g(x,y,\sigma) \vartriangle I(x,y)] \tag{2-27}$$

Amongst the energy functions used, the reduced form of the Mumford-Shah function, that incorporates three tuning parameters (α, λ_I and λ_O) for the evolution of the level set ϕ(x,y), is quite general (Cheng and Rajapakse 2009):

$$\frac{\partial\phi(x,y)}{\partial t} = \left(\alpha\nabla\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|} - \lambda_I(I(x,y)-\mu_I)^2 + \lambda_O(I(x,y)-\mu_O)^2\right)\delta\big(\phi(x,y)\big) \tag{2-28}$$

Where μ_I and μ_O are respectively the mean intensity of the pixels in I(x,y) inside and outside the boundary. The first term smoothens the contour (curvature) as the other two terms drive the contour towards the actual object boundary (Cheng and Rajapakse 2009).

Since images can incorporate a large number of complex objects, several methods computing local energies have been developed. To have a review of these energy functions, the readers are referred to Lankton (2009).

2.4.4.1.2 Applications

There are numerous applications of active contours and level set methods in microscopy. For example, these methods were used to count and characterize myocytes (Acton, Yang et al. 2009), cancer cells (Said, Karam et al. 2007; Ersoy, Bunyak et al. 2008) and neutrophils (Min, Shengyong et al. 2009).

Active contours and level set methods are frequently used in combination with other image processing methods to achieve good segmentation and tracking performances in PCM. For example, a pose-shape manifold algorithm combined with an active contours method was used to segment myocytes and characterize their shape (Basu and Acton 2009). An active contours segmentation method combined with a PCA outlier detection algorithm (to improve robustness, and have a higher segmentation accuracy) was also successful for segmenting leukocytes (Saha, Ray et al. 2009). Active contours were also used to refine the segmentation of cells after pre-processing steps (Schilling, Miroslaw et al. 2007; Bradhurst, Boles et al. 2008).

One of the main challenges when using active contours and level set methods is to have a good initial contour (initialization) that will ensure a proper convergence (usually seeding is required). A level set method was used to detect stem cells boundaries after seeding (cell centers identified) using a criteria to avoid crossing of different boundaries (Huiming, Xiaobo et al. 2009). To segment cells in brightfield images, Ali et al. (2007) used a level set contour that was initialized on a defocused image (with an improved intensity contrast) and that was evolved towards the correct boundary using images of improving focus.

Phase contrast microscopy images containing cells are difficult to segment with active contours and level sets because of the low contrast between the cells and the background and because it is difficult to extract accurately the contour of the cells surrounded by a halo (Orikawa and Tanaka 2010). In fact, the intense and irregular halo surrounding the cells in PCM images causes the edge indicator function (gradient) to pull the level set towards the halo boundary at the exterior of the cells (Ambuhl, Brepsant et al. 2012). Despite these challenges, active contours and level sets are often used to perform or refine cell segmentation in phase contrast microscopy images. For example, cell candidates were identified in PCM using an adaptive rough segmentation (for seeding) followed by a level set method to evolve the contour of the seeds to the actual cell contours, allowing cell

division but prohibiting cell fusion (from a cell segmentation perspective) using constraints (Kang, Miller et al. 2006). An advanced level-set method was used to segment neural stem cells and progenitors in PCM images (Xiong, Chia et al. 2011). Active contours were used to segment and track *Entamoeba histolytica* in PCM images (Zimmer, Labruyere et al. 2002). Level set segmentation was also used to characterize the movements of the contours of keratocyte cells in high resolution phase contrast microscopy to reveal cell edge dynamics (Ambuhl, Brepsant et al. 2012), proposing an energy function that would give better results for processing PCM images. To partly solve the problems associated with the halo surrounding the cells when using level sets, a top-hat transform was used, subtracting the morphological closing of the image from the original image itself (using a disk-shaped structuring element with a radius about the width of the halo) during an intermediate phase of energy minimization (Ambuhl, Brepsant et al. 2012).

2.4.4.2 Region detection methods

2.4.4.2.1 Morphological operations

Morphological operations are basic and powerful tools for extracting image components that are useful for the representation of a region's shape (Gonzalez and Woods 2008). Morphological operations are based on mathematical set theory, and can be used on binary images (black and white pixels) or on grey scale images (Gonzalez and Woods 2008). Morphological operations require a structuring element (SE), which is a small set or subimage used to probe an image under study (closely related to spatial convolution) for properties of interest (Gonzalez and Woods 2008). Usually, morphological operations are pre-processing or post-processing steps for more advanced segmentation techniques performed on a binary image (object versus background) or on a grey scale image, as they simplify image data by preserving shape characteristics and eliminating irrelevancies (Haralick, Sternberg et al. 1987). Among morphological operations, it is possible to consider dilation, erosion, opening and closing (Haralick, Sternberg et al. 1987) as well as basic morphological algorithms such as boundary extraction, hole filling, the extraction of connected components, the computation of the convex hull, thinning, thickening, objects skeletons detection and pruning (Gonzalez and Woods 2008).

Morphological operators are often used to separate single cells in clusters in image segmentation, to refine the cellular contours detected and also to filter out objects that are smaller than a certain size (objects disappear under an erosion with a structuring element

of a specific size (Rodenacker and Bengtsson 2003)). As an example, morphological operators (using an octagonal structuring element) were used to segment cells nuclei in stained Pap smear images (Walker, Jackway et al. 1994). Morphological operators (dilation, erosion) were also used to roughly segment clustered cells in PCM images (Ambriz-Colin, Torres-Cisneros et al. 2006). A background correction step and a threshold followed by a series of dilations and erosions was used to identify the contour of yeast cells in microscopy images (Yu, Elbuken et al. 2011). For detecting cells in cytological material images, Rodenacker and Bengtsson (2003) proposed the following approach: the images are thresholded, the holes inside the mask of the cells are filled and borders of touching cells are deleted and cleaned by a series of opening and closing operations while cells that are too small are filtered out by a series of dilations and erosions (all morphological operations are performed with an octagonal structuring element of a given radius).

2.4.4.2.2 Region splitting and growing methods

The merge and split segmentation algorithm (or "quad-tree" segmentation) was first introduced by Horowitz and Pavlidis (1976), and consists of splitting and merging regions in an image based on an homogeneity or similarity criterion. The algorithm starts with a node that encompasses the whole image, and splits the image into 4 regions (usually squares (Saad, Abu-Bakar et al. 2010)) based on the homogeneity of the regions. This procedure is repeated, and after each splitting step adjacent regions are merged if a similarity criterion is met. This procedure is done until there are no changes in the segmentation result (Liow and Pavlidis 1988). The quad-tree method was used to segment histological plant cells, white blood cells, and red blood cells in microscopic images (ByoungChul, MiSuk et al. 2007). This technique was also used to segment brain lesions in MRI images (Saad, Abu-Bakar et al. 2010).

In region growing segmentation, the first step is to highlight manually or by automatic methods (thresholding or other techniques) a region of interest in each object to be segmented (referred as "seeds") (Adams and Bischof 1994). The seed pixels are grown to their neighbor's pixels and those neighbor pixels will be included only if they satisfy specific conditions (Sulaiman, Isa et al. 2010). Usually potential pixels surrounding the regions are compared to the mean intensity of the pixels included in each region (Adams and Bischof 1994). This process is repeated until no more pixels can the added to each

region. There are several applications of region growing segmentation methods for processing cellular microscopic images. It is commonly used to segment the nuclei and the cytoplasm of stained cell images in fluorescence microscopy (using fluorophores) or brightfield microscopy (histological samples with dyes). This techniques was used to distinguish overlapping cervical cells in PAP tests color images (Sulaiman, Isa et al. 2010). The same technique was also applied to pseudo color images for PAP tests diagnostics. Pseudo color is a method of assigning colors to grey values according to a table of function (Sulaiman, Isa et al. 2010). A seeded region growing method (automatic seeding generated by a threshold) was used to generate blob features (avoiding exact cell segmentation) that were fed to a Support Vector Machine (SVM) classifier in order to detect mouse mesenchymal stem cells undergoing mitosis in PCM images (Anan, Kang et al. 2010). Mitosis detection is based on the fact that cells undergoing mitosis have a drastic morphological change, both daughter cells having an “8” shape (Anan, Kang et al. 2010). To detect cells undergoing mitosis, a region growing algorithm started (seeded) and stopped using two different thresholding techniques was used in PCM (Liu, Li et al. 2011).

2.4.4.2.3 Nested kernels

Nested kernels constitute a special case of region detection method, which are based on the computation of a local pixel pattern around the cells centroid (for example a Gaussian distribution of the grey level values). The goal is not to segment exactly cellular areas, but to give an indication of where the cells are for tracking purposes (track cells centroids). Nested kernels are increasingly used for cell tracking via a mean shift strategy. Since in phase contrast microscopy cells have a darker interior with a white halo around them, nested kernels can be roughly fitted on the region corresponding to the interior of a cell starting from its centroid. For example, a mean-shift procedure using 3 nested kernels (one attracted by the inner dark pixels, one attracted by the white halo pixels and one attracted by the external dark pixels) was used to roughly segment and track different kinds of cells in PCM (Debeir, Van Ham et al. 2005). Other kernels (oriented ellipse kernel, non-Gaussian kernel and non-Gaussian ellipse kernel) were proposed (Xiaodong, Houqiang et al. 2006) to capture areas associated with cells in fluorescence microscopy (more robust to changes in cells shape and orientation), and these kernels where used for cell tracking via a mean-shift process. Also, a Gaussian kernel was used to enumerate fluorescent spots in fluorescence microscopy (Selinummi, Ruusuvuori et al. 2009).

2.4.4.2.4 Watershed transform

A classical method (that could be classified as a region growing algorithm) is the watershed transform, for which an image is compared to a topographical map in which minima are considered as low points and maxima as high points (Vincent and Soille 1991), and as water accumulates in low points, divisions between objects can be identified and regions are highlighted. A good literature review on the watershed transform technique is provided by Quanli et al. (2010). In fact, any grey scale image can be considered as a topographical surface or landscape, and the original watershed transform is a segmentation method that simulates a landscape being flooded by rain in which water will flow to some regional minima (Quanli 2010). Watershed lines that separate catchment basins are treated as boundaries between regions for the purpose of segmentation. An alternative model is based on an immersion simulation introduced by Vincent and Soille (1991), in which the surface is flooded from its regional minima and the merging of water from different sources is prevented by building dams, as illustrated in Figure 2-28.

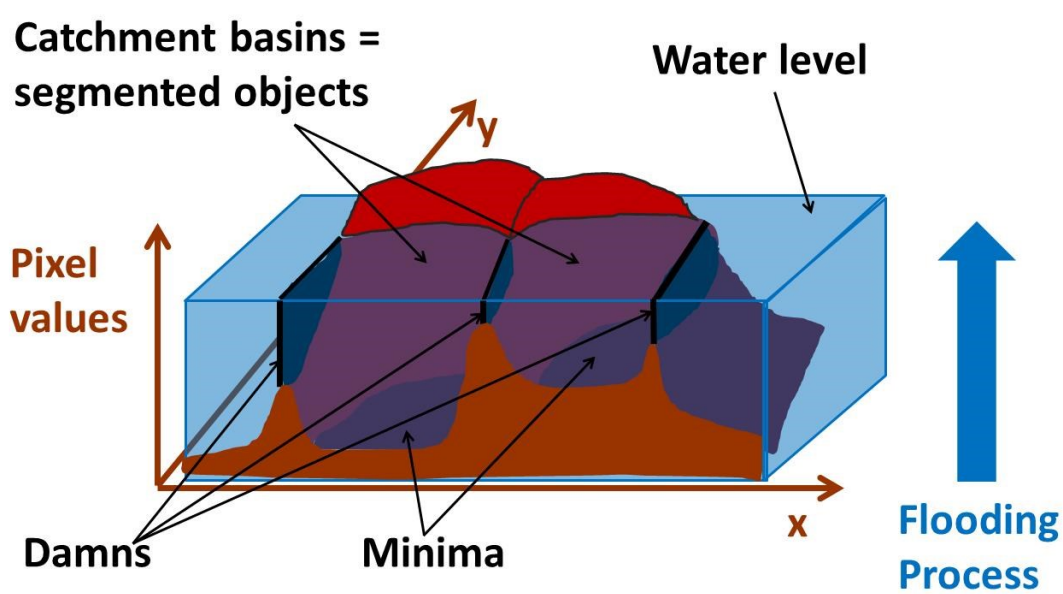


Figure 2-28 : Watershed algorithm approach: as water fills catchment basins, pixels in an image are grouped together, starting from minima, until two groups of pixels reach each other, thus detecting a watershed line (dam) (Vincent and Soille 1991).

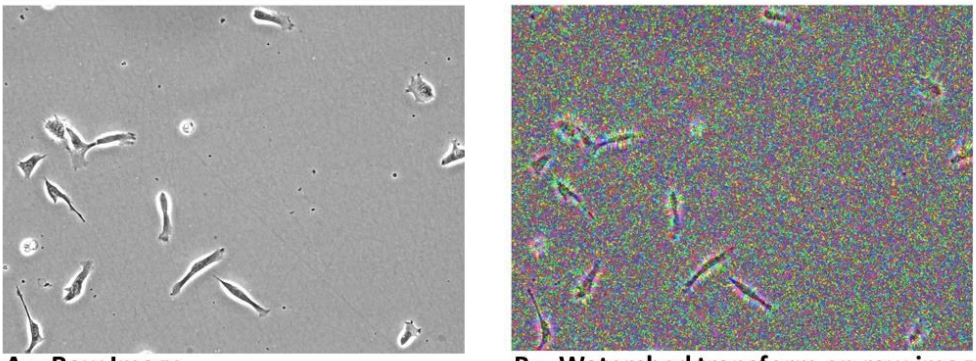
The image is thus partitioned into two different sets: catchment basins and watershed lines (dams), which result in a segmentation of the original image (Quanli 2010). The regions surrounded by watershed lines represent the set of pixels reachable by monotone

decreasing or increasing distance from the local maximum or minimum grey scale value (Rodenacker and Bengtsson 2003). The immersion method is based roughly on a sorting of the pixels in increasing order of their grey values, and on a “fast breadth-first scan” of the “plateaus” enabled by a first-in-first-out type data structure (Vincent and Soille 1991). Even though the watershed transform is commonly used for image segmentation, it often leads to over-segmentation due to local minima. To reduce the effect of over-segmentation, as a pre-processing step, the H-minima transform is often used to prevent over-segmentation by suppressing all minima less than a particular depth (h) from the local pixel values (Cheng and Rajapakse 2009). A complementary algorithm to the watershed transform is the immersion simulation based self-organizing (ISSO) transform, which allows users to define a self-organizing function to model how the objects of interest should self-organize based on *a priori* information such as size, shape or border smoothness during the immersion process (Quanli 2010).

In order to solve the over-segmentation issue, there are mainly two solutions: hierarchical (fragment merging) or marker-controlled watershed methods (Xiaodong, Houqiang et al. 2006). Marker-controlled and hierarchical methods are often used to get rid of the over-segmentation problem by providing strong prior information to preprocess the image before using the watershed transform, or by applying merging rules to post-process over-segmented regions (Quanli 2010). In hierarchical methods, it is possible to use a connectivity-based approach (which might fail if the fragments are too big) or a shape and size-based approach (which is more effective, but still needs to be refined by post-processing steps) (Xiaodong, Houqiang et al. 2006). When using marker-controlled watershed, minima are imposed by identifying predefined markers (Xiaodong, Houqiang et al. 2006), and this operation is usually referred to as “seeding”. For marker-controlled watershed to work and avoid over-segmentation, it is necessary to ensure that there is only one marker per object, thus making marker extraction a difficult task (Xiaodong, Houqiang et al. 2006). A marker is a set of connected pixels in an image which represents the existence of an object (Xiaodong, Houqiang et al. 2006). In a marker-controlled approach, flooding begins from the objects markers (imposed as global minima) instead of the regional minima. Since each marker represents one object, it is important to extract them correctly to avoid over-segmentation or under-segmentation (Xiaodong, Houqiang et al. 2006).

The segmentation performance of the watershed transform not only depends on the markers used but also on the marking function (the topographic surface flooded by water) computed from the binary image containing the markers, which should synthesize physical characteristics of the objects to segment (Fatima and Seenivasagam 2011). In order to properly identify the number of objects to be segmented, a marking function is usually used (Vincent and Soille 1991). In most cases, the inner or the outer distance transform is used on the binary image containing the markers to generate a marking function for the watershed transform (Cheng and Rajapakse 2009). For separation of partially overlapping objects in binary images, the inverse inner distance map is usually chosen as a classical marking function (Fatima and Seenivasagam 2011). For segmenting clustered cell nuclei, the outer inner distance map computing the distance from shape markers can be used, avoiding jagged boundaries of segmented objects (Fatima and Seenivasagam 2011). From a computational perspective, the outer distance transform converts a binary image into a distance map where every background pixel has a value corresponding to the minimum distance from shape markers (Fatima and Seenivasagam 2011). In comparison, the inner distance transform converts a binary image consisting of foreground and background pixels into a distance map in which each foreground pixel has a value corresponding to the minimum distance from the background (Cheng and Rajapakse 2009).

Figure 2-29 illustrates how a typical PCM raw image (A) can generate a significant over-segmentation (B) when subjected to a watershed transform because of local minima (each colored region corresponding to one cell detected):



A – Raw Image

B – Watershed transform on raw image (oversegmentation)

Figure 2-29 : A-Raw image; B-Watershed transform applied directly on a raw image. There is significant over-segmentation, with a large number of tiny objects segmented (one color for each group of pixel segmented).

In order to avoid over-segmentation, using markers is almost mandatory, as illustrated in Figure 2-30. For segmenting a raw image (A), cellular regions are identified (B) by thresholding a range map of the image and after by filtering out cellular pixel groups smaller than a certain size (filter size of 5 pixels, threshold of 20 and minimum number of pixels of 5000). To compute cell markers (C), pixels having mid-range grey level values associated with regions inside cells (between 140 and 235) and linked to pixel groups larger than 500 pixels are highlighted and considered as object markers (red). To compute a background marker (C), the watershed lines identified by computing the watershed transform of the background binary matrix (inverse of cellular regions binary matrix computed in B) are used (red). In the background binary matrix, global minima are imposed for the pixels corresponding to object and background markers, and the watershed transform provides the segmentation result showed in D, for which there is one color per cell.

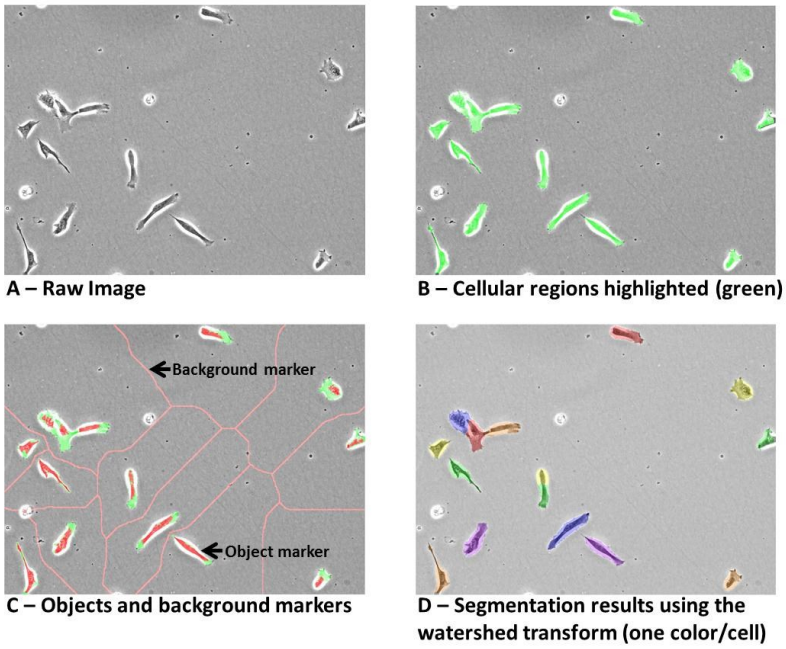


Figure 2-30 : A-Raw image; B-Cellular regions highlighted (green) for object markers; C-Background and object markers identified (red); D- Segmentation results using the watershed transform (one color for each cell).

2.4.4.2.5 Applications

The watershed transform is commonly used to segment stained cells in fluorescence microscopy (fluorophores) and in brightfield microscopy (using dyes) since the fluorophores and the dyes used can highlight specific regions inside the cells, thus providing ideal object markers. Numerous examples can be found in the literature. For example, a watershed method using markers generated by a series of morphological operators was used to segment and track cancer cells to study their cycle progression in fluorescence microscopy (Xiaodong, Houqiang et al. 2006): the seeds obtained by thresholding were refined using coarse and fine erosion structures through morphological operations (Xiaodong, Houqiang et al. 2006). Clustered nuclei in fluorescence microscopy images were segmented and separated by a rough segmentation using active contours (to generate seeds), by computing the inner distance map for each cluster detected, by identifying regional minima (by thresholding) and by using the H-minima transform (Fatima and Seenivasagam 2011). In fluorescence images of mouse neurons and Drosophila cells, nuclei were first roughly segmented by an active contour algorithm to generate seeds which were in turn used to segment whole cells using a watershed-like algorithm (Cheng

and Rajapakse 2009). A watershed method was used to segment stained cells for cancer diagnosis (Lu, Bai et al. 2008) relying on thresholding and edge detection (Gaussian coefficients gradient) to reduce over-segmentation.

Using the watershed transform for segmenting unstained cells in PCM, DIC and brightfield microscopy is more challenging, since there are no molecular markers (fluorophores or dyes) to highlight specific regions inside the cells, making the identification of object markers more difficult. Despite this challenge, several authors developed methods to generate seeds in order to use the watershed transform while avoiding over-segmentation. For example, a seeded constrained watershed transform using a convolution between the image and a typical cell template (in order to identify seeds) was used to segment hematopoietic stem cells in DIC images (Kachouie, Fieguth et al. 2010), and performed well for round cells. Local features (such as variance) were used to generate foreground and background seeds to segment individual neural stem cells in clusters by the watershed transform applied to brightfield images (Qian, Peng et al. 2009). To segment neural stem cells in adherent culture conditions (brightfield microscopy), an hybrid image segmentation framework based on a self-organizing map and the seeded watershed algorithm was used (Xiang and Datian 2009). A watershed algorithm (combined with a growing region approach) with markers generated by a combination of a threshold and the distance transform was used to segment HeLa cells in PCM images (Orikawa and Tanaka 2010). The watershed transform was used to segment different human cellular phenotypes in PCM coupled to holography (Mõlder, Sebesta et al. 2008).

2.4.5 Pattern recognition

2.4.5.1 Features definition

In image processing, a *feature* is a quantitative value that can be used for pattern recognition. A pattern is an arrangement of descriptors or features that can be used to classify pixels within an image for segmentation (Gonzalez and Woods 2008) or for classification purposes. A good review of the different types of features used to study biological samples (cytometry) was provided by Rodenacker and Bengtsson (2003). When an image is analyzed, both the measured grey values of the pixels and the spatial arrangements of these values can be studied to generate numerical values or features that carry some information of interest about the imaged objects (Rodenacker and Bengtsson 2003).

The feature generation process is illustrated in Figure 2-31. From a raw digital image, the selection of the measurement zone delimits the region to be quantified (e.g. cells nucleus, cytoplasm of whole cells, etc.) and requires some sort of segmentation that produces a map to count and localize the objects of interest (Rodenacker and Bengtsson 2003). It is also possible to perform a mathematical transformation of the raw digital image to highlight certain aspects of the distribution of pixel values (e.g. thresholding, grey level co-occurrence matrix, Fourier Transform, etc.), thus producing a data array corresponding to a new image, a matrix of co-occurrence values or a frequency distribution (Rodenacker and Bengtsson 2003). The final step is a measurement process that produces a feature value (scalar) corresponding to a count, an integration or a selection performed on a segmentation map and/or a transformation data array (Rodenacker and Bengtsson 2003). For example, if a transformation performed on an image results in a frequency distribution (histogram), the distribution associated to this histogram can be characterized by several measurements (moments) such as an average, a standard deviation, a skew and a kurtosis (Rodenacker and Bengtsson 2003).

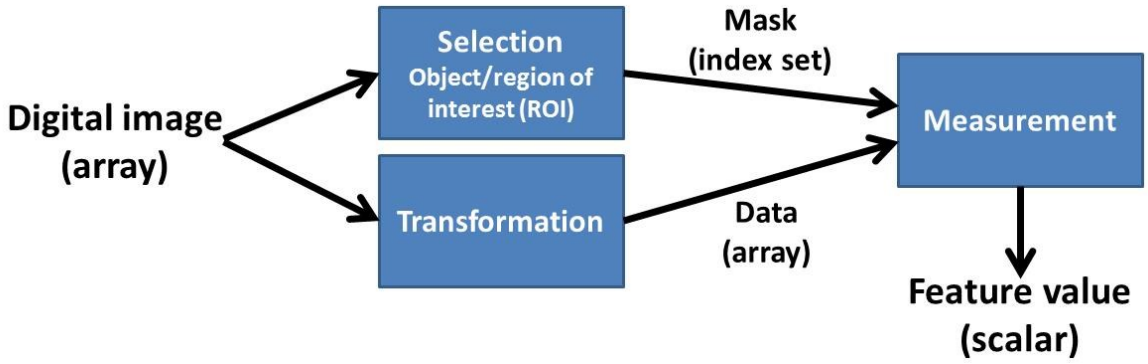


Figure 2-31 : Feature generation process (adapted from Rodenacker and Bengtsson (2003))

As morphometric features express only the spatial arrangement of grey pixel values, densitometric features look at overall grey values without considering spatial distributions (Rodenacker and Bengtsson 2003). Textural and structural features are a combination of both since they consider both the spatial distribution of the grey values and the overall grey values (Rodenacker and Bengtsson 2003). As illustrated in Figure 2-32, features can be classified as intensity features, textural/structural features and shape/morphological

features (Rodenacker and Bengtsson 2003). The following sections provide more details about each type of features.

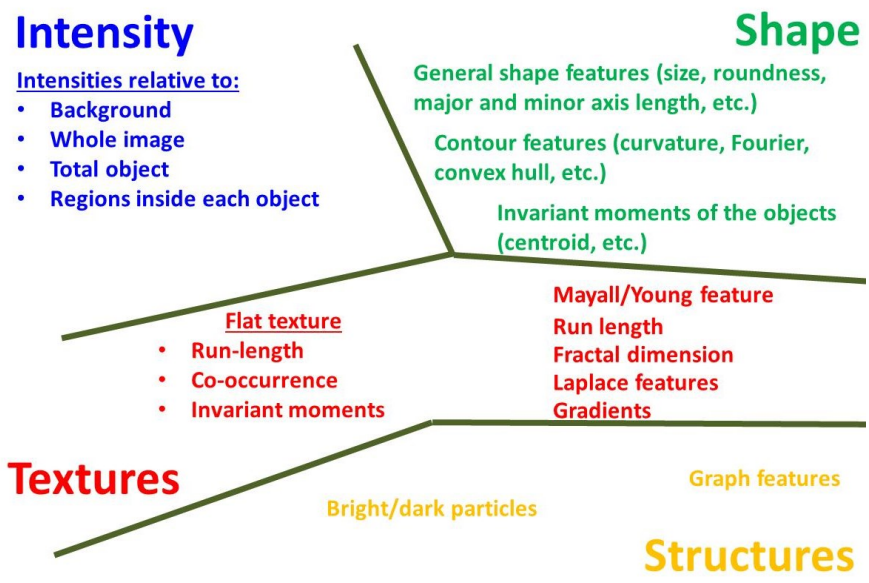


Figure 2-32 : Different classes of features (adapted from Rodenacker and Bengtsson (2003))

It is important to note that features which are directly extracted from the raw image (primary features) can be combined to form new features (secondary features) that can better express the desired properties of the objects to be quantified (Rodenacker and Bengtsson 2003). In this PhD project, it was necessary to select the most appropriate features to characterize cellular clusters and individual cells imaged in PCM.

2.4.5.2 Features selection

There is almost an infinite number of features (or combination of features) that can be computed from a single image. Because of the huge number of features that can be generated and because they are often highly correlated, singular value decomposition (SVD) or principal component analysis (PCA) can be used to reduce the dimensionality of the features space (Jun, Cooper et al. 2010; Skoczylas, Rakowski et al. 2011). For example, a features space was reduced from 295 features to 25 latent variables using singular value decomposition before using a Support Vector Machine (SVM) classifier for performing unstained viable cell recognition in phase contrast microscopy (Skoczylas, Rakowski et al. 2011). A principal component analysis (PCA) was used to capture most of

the variance in a features space before feeding the information to a neural network (Qian, Peng et al. 2009; Xiang and Datian 2009).

Some features are critical for classification as other features are less important, and non-significant features might be removed for computational efficiency. Many methods are described in the literature: bottom-up approaches start with an empty feature set and include features incrementally in an optimal manner while top-down approaches include at first all features and remove the ones that provide redundant information (Schilling, Miroslaw et al. 2007). For example, for performing cervical cancer diagnosis from texture analysis in PCM images, optimal features were selected via a sequential forward floating selection (SFFS) method considering as an objective function the classification error rate, the Bhattacharyya distance and scatter matrices (Schilling, Miroslaw et al. 2007).

2.4.5.3 Intensity features

The simplest feature is the intensity level of a pixel (grey value for monochrome images). Intensity features (densitometric) rely on the absolute values of the intensity measurements in an image, and usually do not consider the spatial position of individual pixels (Rodenacker and Bengtsson 2003). These features are based on the histogram of pixel intensities, from which pixels can be classified easily using thresholding techniques. However, after a rough segmentation of the image, the local pixel intensities can be studied, as it is possible to obtain the histogram of pixel grey level values for the whole image or subsets of the image such as the background or the zone inside an object (Rodenacker and Bengtsson 2003).

In pattern recognition, when dealing with a series of images having the same size in terms of pixels, each pixel location can be considered as a feature having as a value its grey level intensity. When using the Karhunen-Loève expansion (or better known as principal component analysis or PCA) for pattern recognition (applied on an image dataset), each pixel in each image (size m x n) is considered as a single occurrence in a m*n feature space. This method is commonly used in human facial recognition. For an image dataset of K images, it is possible to obtain a data matrix of K rows by m*n columns. Each pixel intensity is corrected for the mean value of the whole image dataset (Sirovich and Kirby 1987), referred to as “mean image”. It is possible to capture most of the information with several components (Turk and Pentland 1991). The first step is apply PCA analysis on a training dataset and keep only the components capturing most of the variance (“C”

components). The dot products of each image with each training score image ("eigenfaces" or "eigencells") can be used as features, giving a total of "C" features for each image (Turk 1991). These features can be assigned to a number of classes. A common method to classify an image is to compute the Euclidian distance between the "C" features of a new image and the typical "C" features of each class, and to assign the image to the group having the shortest distance. Another method to classify a new image is to compute and minimize the Euclidian distances between the raw image (from which the mean image is subtracted) and the projections of the raw image using each set of eigenvectors associated to each class. Several applications are reported in the literature. For example, fluorescence images of lymphocytes were decomposed into smaller image patches, and these patches were classified using PCA generated features fed to a neural network as cells or non-cells patches (Nattkemper, Ritter et al. 2001). The PCA/neural network cell detection technique was combined with a features binding algorithm (a recurrent neural network) in order to identify cell contours (Nattkemper, Wersing et al. 2002). Similar results were obtained using independent component analysis (ICA) techniques (Nattkemper, Twellmann et al. 2003). A variant of the PCA method to generate features is the Fisher linear discriminant (FLD) analysis of image patches (Long, Cleveland et al. 2005), which was used to identify unstained cells in brightfield images. Analyzing image patches in a training dataset with the FLD method has the advantage of maximizing the ratio of between-class variance to within-class variance, as opposed to PCA which maximizes both the between-class and within-class variances captured (Long, Cleveland et al. 2005). The FLD method combined with a neural network classifier was found to be more robust relative to size, illumination, focus and noise than the PCA method (Long, Cleveland et al. 2005). In another application, the FLD method was also used to classify pixels as background, living cell or necrotic cell (Malpica, Santos et al. 2003).

2.4.5.4 Textural features

Image texture can be defined as "the spatial variation of pixel intensities (grey values)", and can be perceived at different scales or levels of resolution (Tuceryan and Jain 1999). Another author defines textures as "complex visual patterns composed of entities that have specific brightness, color, slope, size and scale, giving rise to perceived brightness, uniformity, coarseness, smoothness and granularity in an image" (Caponetti, Castellano et al. 2009). Texture analysis can be used for a variety of tasks such as image recognition, classification, segmentation, synthesis or compression, and the literature reports several applications such as product inspection, medical image analysis, documents processing

and remote sensing (Tuceryan and Jain 1999). Psychophysical research showed that humans and primates perform a frequency analysis of images in their visual cortex, analyzing textured images by decomposing them into their frequency and orientation components (Tuceryan and Jain 1999). This is why we distinguish textures intuitively without having to think about it.

For industrial and research applications, since the intensity variations in an image which characterize a texture are due mainly to underlying physical variations in a material/scene, texture analysis can be used to model physical variations that are difficult to detect (Tuceryan and Jain 1999). The extraction of textural features can be performed on the whole image or can involve the quantification of the local density variability inside an object of interest after segmentation (Tuceryan and Jain 1999).

Textural analysis methods can be classified into four categories: statistical methods, geometrical methods, model-based methods and signal processing methods (Tuceryan and Jain 1999; Liu 2004).

2.4.5.4.1 Statistical methods

Statistical methods describe the texture of a region based on the spatial distribution of grey level values by generating features based on statistical moments (Tuceryan and Jain 1999; Liu 2004). The most popular method is the grey level co-occurrence matrix (GLCM), in which a matrix is produced to count the number of grey value pairs associated with a certain displacement vector (Tuceryan and Jain 1999). For each displacement vector, a square matrix having a size equivalent to the number of grey values (L) is computed, as illustrated in Figure 2-33:

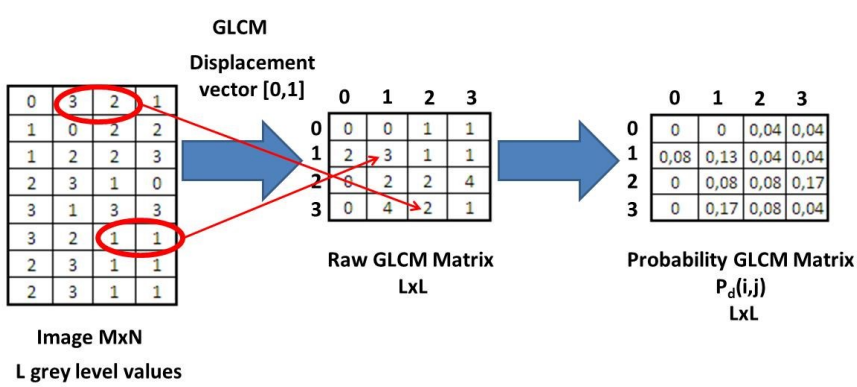


Figure 2-33 : GLCM technique

In Figure 2-33, the raw GLCM Matrix is computed by looking at pairs of grey level values that are linked by a displacement vector. Each pixel grey level value in the image is associated (as a pair) to the pixel grey level value that is 0 pixel down and 1 pixel to the right (displacement vector [0,1], corresponding to a distance of 1 pixel in the 0 degree direction). The raw GLCM Matrix is then built by computing the number of occurrences for each possible grey level value pair. It is possible to compute raw GLCM Matrices for several displacement vectors (different distances and different angles). From the raw GLCM Matrix, a Probability GLCM matrix can be computed by dividing each count by the total number of counts (Figure 2-33). For each Probability GLCM matrix (for different displacement vectors), several scalar features can be calculated such as the maximum probability, correlation, contrast, uniformity (energy), homogeneity and entropy (Haralick, Haralick et al. 1973).

If the texture is homogenous (no directionality), as it is usually the case for chromatin distribution inside cells nuclei, the number of displacement vectors used can be limited to orientations of 0 and 90 degrees (Rodenacker and Bengtsson 2003). However, if significant directionality is expected, several orientations should be studied through different displacement vectors, and the maximum, minimum and range values for each raw GLCM Matrix generated should be used to express the directional inhomogeneity of the texture (Rodenacker and Bengtsson 2003).

Several applications using GLCM features for analyzing biological samples are described in the literature. For example, texture analysis using GLCM features was used to classify Pap Smear cells nuclei as normal or abnormal in stained images (Walker, Jackway et al. 1994). Also, to classify cells as healthy, apoptotic and necrotic in dark field micrographs from an *in situ* microscope, textural features (GLCM and other features) were used on image patches (roughly the same size than the cells), the classification being performed by a SVM classifier (Burgemeister, Nattkemper et al. 2010).

2.4.5.4.2 Geometrical methods

Geometrical methods characterize a texture as being composed of textural elements or primitives having a defined geometry (Tuceryan and Jain 1999). Once textural elements in an image are identified, it is possible to either compute statistical properties from the identified textural elements or to extract a placement rule that defines a particular texture

(Tuceryan and Jain 1999). Several methods rely on Voronoi polygons, which are constructed by tessellation from texture tokens (points of interest in an image) (Tuceryan and Jain 1999).

2.4.5.4.3 Model-based methods

Model-based methods rely on the construction of an image model that can be used to describe or synthetize a texture, as model parameters capture the perceived texture quality (Tuceryan and Jain 1999). The most popular methods are autoregressive models, Markov Random Fields (MRF) and fractal models (Tuceryan and Jain 1999; Liu 2004).

2.4.5.4.4 Signal processing methods

Signal processing methods based on transforms convert an image into a new domain (change in coordinate system) to analyze spatial frequencies generated by the variation of pixels intensities (Tuceryan and Jain 1999; Liu 2004). These transforms usually rely on spatial or frequency filters. Some specialized transforms are also reported in the literature and can be focused specifically on biological applications. For example, the Mayall/Young Chromatin feature transform is a special transform taking the global shape of the object (chromatin distribution in cells nuclei) into account to generate textural features such as heterogeneity, clumpness, etc. (Rodenacker and Bengtsson 2003). Another interesting transform mostly used to analyze the texture of powders (isotropic) is the Angle Measure Technique (AMT), in which an image is unfolded into a 1D vector and in which the mean angle and the mean difference is computed between consecutive points over several scales (Andrle 1994; Mortensen and Esbensen 2005).

Spatial filters that are edge detectors are appropriate to differentiate fine textures from coarse textures, since fine textures will have a higher density of edges per unit area (Tuceryan and Jain 1999) or a higher local rate of change in pixel values. For example, gradient filters generate features that quantify the velocity of change in grey scale values. Laplacian filters correspond to the sum of partial local derivatives, and the corresponding features matrix delivers a certain description of changes in pixels neighborhoods (Rodenacker and Bengtsson 2003). Amongst gradient/edge detector filters, it is also possible to consider non-linear filters such as the standard deviation filter and the range filter and also advanced edge detectors such as the Canny and the Sobel filters (Gonzalez and Woods 2008).

Frequency filters rely on the spectral analysis of an image to generate textural features. A first approach to generate textural features is to divide the Fourier Transform of an image (FFT, frequency domain) into rings (frequency content) and wedges (orientation content) (Tuceryan and Jain 1999), as shown in Figure 2-34.

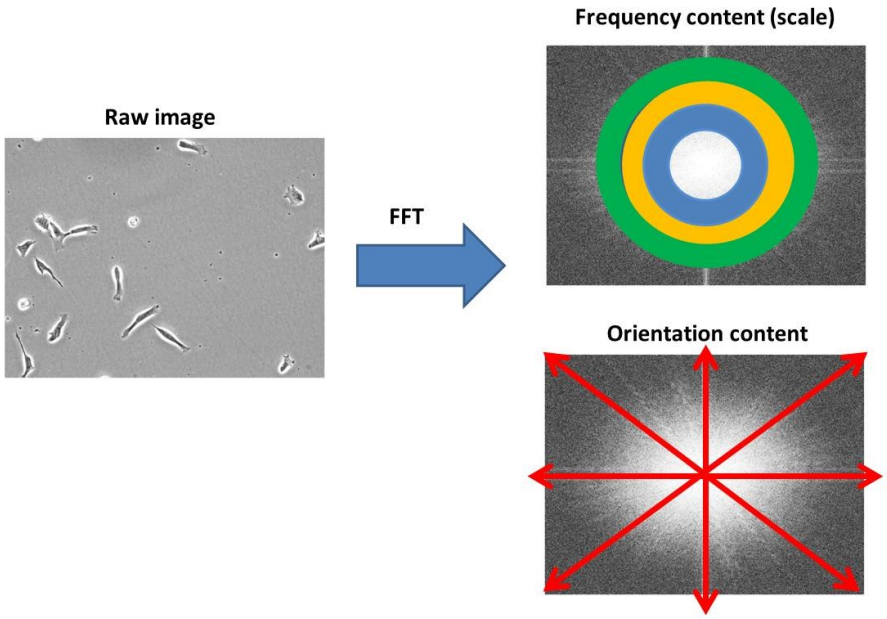


Figure 2-34 : Frequency content (scale) and orientation content information in the FFT of an image

To generate features, the FFT transform of an image (frequency domain) is divided into regions, and the total energy in each region is representative of the scale/orientation of the textural information (Tuceryan and Jain 1999). The energy computed in each shaded band (e.g. green, yellow and blue bands in Figure 2-34) is a textural feature related to the coarseness/fineness (scale) information and the energy computed in each wedge (red arrows from the center of the FFT) is a textural feature related to directionality information (Tuceryan and Jain 1999).

The wavelet transform is preferred to the Fourier transform since it looks at both the frequency and spatial information within an image (Liu 2004; Gonzalez and Woods 2008). Since wavelets are of particular interest for this thesis, they will be detailed further in a dedicated section.

2.4.5.5 Shape-based (morphometric) features

Shape-based or morphometric features express the overall size and shape of an object (or a cell) in an image (Rodenacker and Bengtsson 2003). Several features rely on a segmentation of the cells to be performed first (segmentation shape-based features) as other methods rely on a relative fit of an object/cell template in the image (fit shape-based features).

2.4.5.5.1 Segmentation shape-based features

Segmentation shape-based features are computed after a segmentation step is completed, in which each object O_i and each border C_i is identified via a segmentation map, as illustrated in Figure 2-26 (Rodenacker and Bengtsson 2003). A straightforward way to estimate the size of an object in an image is to find the minimum and maximum x and y coordinates associated with the object mask, thus defining a bounding box representative of the height, width and area of the object (Rodenacker and Bengtsson 2003). The ratio of the area of the actual object to the bounding box area provides a simple shape factor feature (Rodenacker and Bengtsson 2003). Once a formal segmentation of the cells is achieved, it is possible to compute geometrical features such as cell area, cell perimeter, cell orientation or cell eccentricity (Kachouie, Fieguth et al. 2010). Geometric moments can also be computed, but they can be sensitive to the overall cell size and orientation (Rodenacker and Bengtsson 2003). In Matlab®, it is possible to compute several segmentation shape-based features such as the area, the centroid coordinates, the major axis length, the minor axis length, the orientation, the equivalent diameter and the perimeter using the *bwlabeln* and *regionprops* functions, as illustrated in Figure 2-35.

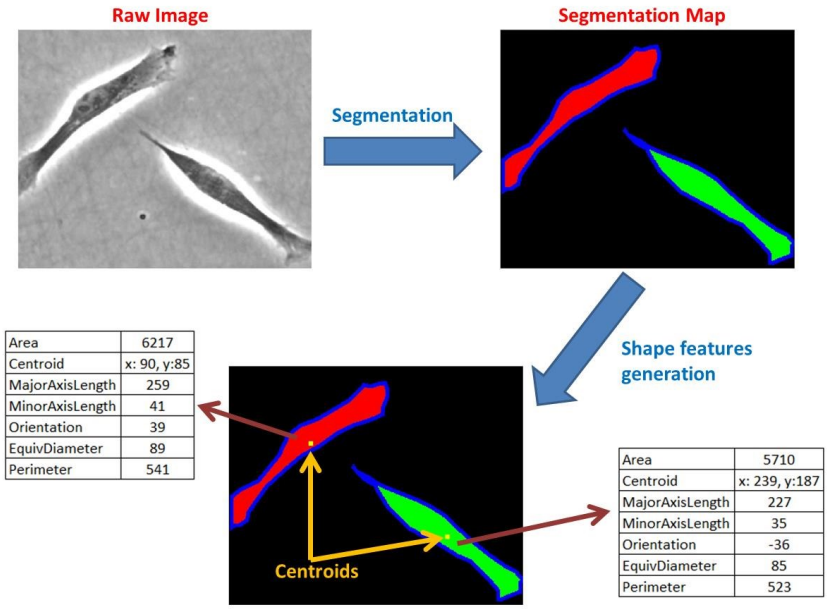


Figure 2-35 : Shape-based features generated from a segmentation map

Many segmentation shape-based features are used in the literature as a segmentation post-processing step to refine segmentation results (e.g. remove or add pixel groups in the segmentation map according to their size/area) or to characterize different culture/process conditions (e.g. cells grow bigger in condition 1 than in condition 2). For example, geometrical/shape-based features were used to distinguish yeast cells depending of their status in their cell cycle (Yu, Elbuken et al. 2011). Also, House et al. (2009) used a cut size to remove clutters (groups of pixels that are too small) after a rough segmentation. Shape and textural features (among them Fourier descriptors) were used to detect cells in an image, approximating the cellular cytoplasms using a Voronoi diagram (Skoczylas, Rakowski et al. 2011). In order to classify differentiated and non-differentiated neural stem cells, Fourier descriptors were used to characterize their shape after single cell segmentation (Xiang and Datian 2009). Also, to locate the center of mass of artificial phospholipid vesicules in PCM images, Usenik et al. (2011) used a system of two equations in polar coordinates assuming that the vesicules were round. Finally, to classify fibroblast cells according to their morphology (spread, polarized, oriented) in PCM images, a set of 18 segmentation shape-based and 21 appearance-based features were used (Theriault, Walker et al. 2012). The segmentation shape-based features were calculated from a segmentation map generated by a multi-scale approach in which the magnitude of

the gradient and the Laplacian were computed at each scale (Theriault, Walker et al. 2012).

2.4.5.5.2 Fit shape-based features

Fit shape-based features rely on methods that try to fit known patterns (cells having a certain shape for example) in the image after edges have been highlighted by an edge detection method or by matching a region profile by convolution (probability map generated with maxima corresponding to cells location). This may be referred to as *a priori* knowledge of the cells shape (Caponetti, Castellano et al. 2009).

For cell localization, it is possible to convolve the image with a typical template of a cell (used as a filter) in order to generate a probability map that can be thresholded to locate cells (Kachouie, Fieguth et al. 2007; Gonzalez and Woods 2008; Kachouie, Fieguth et al. 2008; Kachouie, Fieguth et al. 2010). In a similar fashion, ring filters of different diameters can be used to detect round cells (e.g. hematopoietic stem cells) in PCM images (Sungeun, Bise et al. 2010). The method consists of convolving the image with ring filters of different radii and to threshold the output (fitted with a quadratic surface to remove noise) to identify local maxima (pixels with the highest probability of being a cell candidate) (Sungeun, Bise et al. 2010). In each case, the probability/correlation map can be considered as a matrix of features that needs to be processed by a classifier (usually thresholding) to identify cells. This technique works very well to locate round cells centers (in this case myoblasts not adhered yet, which are round), as shown in Figure 2-36:

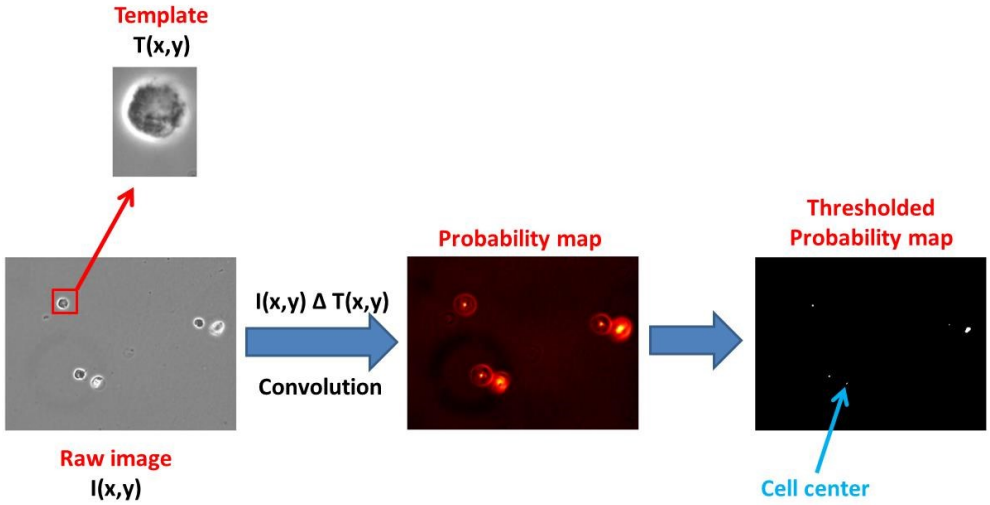


Figure 2-36 : Round cells localized by template convolution

Another method is to obtain a probability map for the localization of the cells by computing the probability that each pixel belongs to a round or elliptic object of a specific radius or geometry (Kachouie, Lee et al. 2005; Kachouie, Fieguth et al. 2007). This probability map can be thresholded to localize the cells and get shape information (Kachouie, Lee et al. 2005; Nezamoddin, Paul et al. 2006). Before fitting circles or ellipses, it is usually necessary to highlight key points in the image through the use of a transform such as an edge detector (Kachouie, Fieguth et al. 2010) or the use of the Hough transform (Demirkaya, Asyali et al. 2009). The Hough transform is a powerful technique which can be used to isolate features of a particular shape in an image (usually pixels that are related to object boundaries). This technique can be used to detect regular curves such as lines, circles or ellipses (Caponetti, Castellano et al. 2009). The main advantage of the Hough transform is that it is tolerant of gaps in curve descriptions and is relatively unaffected by image noise (Caponetti, Castellano et al. 2009). For instance, the Hough transform was used to highlight the cytoplasm boundary pixels of oocyte cells after computing the gradient image (Caponetti, Castellano et al. 2009). The Hough transform was used also to detect round cells in *in situ* microscopy (for cellular suspensions) (Burgemeister, Nattkemper et al. 2010). To quantify the size distribution of different species of microalgae (round cells) exposed to several culture media, a 2D Gaussian elliptical pattern was fitted by regression on each cell imaged by a microscope (10X), and the fit parameters served as shape features (McConico, Horton et al. 2012). Finally, circles in an image can be detected using an algorithm that highlights sharp transitions (by using the Hough transform or another edge detector) and measures the degree of fit between the resulting edge map and circles within a definite range of radiuses (Demirkaya, Asyali et al. 2009). The probability map is generated by computing all the pixels belonging to circles of specified radii centered at each edge pixel (Demirkaya, Asyali et al. 2009). Circles related to cells can be identified by thresholding (the highest probabilities being associated to the center of the cells).

2.4.6 Wavelet transforms

Wavelets constitute a multiresolution analysis approach in which a scaling function is used to create a series of approximations of a 1D signal or a 2D signal (an image), each differing in resolution (over several scales). Wavelets are used to capture the difference of information between adjacent approximations (Gonzalez and Woods 2008). Wavelet coefficients capture subband information that corresponds to specific frequency

bandwidths. A good review of the different wavelet transforms available is presented by Starck, Elad et al. (2004) and the concepts are well illustrated by Gonzalez and Woods (2008).

2.4.6.1 1D wavelet transform

For a 1D discrete function f(x) described by *M* data points, a wavelet expansion can be computed using the discrete wavelet transform (DWT) (Gonzalez and Woods 2008):

$$f(x) = \frac{1}{\sqrt{M}} \sum_{k} A(j_0, k)\varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k} W(j, k)\psi_{j,k}(x) \tag{2-29}$$

In which j_0 is the original scale, *j* is the scale number, $\varphi(x)$ is the scaling function (low pass filter), and $\psi(x)$ is the wavelet function (high pass filter). In the wavelet expansion, A(j,k) are approximation coefficients and W(j,k) are detail/wavelet coefficients computed by convolution between the filters and the original signal f(x) (Gonzalez and Woods 2008):

$$A(j_0, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\varphi_{j_0,k}(x) \tag{2-30}$$

$$W(j, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\psi_{j,k}(x) \tag{2-31}$$

The scaling and wavelet functions must be scalable to scale *j*, as illustrated in the following equations for dyadic scaling (Gonzalez and Woods 2008):

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k) \tag{2-32}$$

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \tag{2-33}$$

The scaling and wavelet functions are linked via recursive equations, assuming that the wavelets constitute an orthonormal basis for the DWT expansion (Gonzalez and Woods 2008):

$$\varphi(x) = \sum_{n} h_{\varphi}(n)\sqrt{2}\,\varphi(2x - n) \tag{2-34}$$

$$\psi(x) = \sum_{n} h_{\psi}(n)\sqrt{2}\,\varphi(2x - n) \tag{2-35}$$

$$h_\psi(n) = (-1)^n h_\varphi(1-n) \tag{2-36}$$

Where h_φ is called the scaling vector and h_ψ is called the wavelet vector. The wavelet vector and the scaling vector can be used directly in a multistage structure to compute the wavelet coefficients by convolution in an efficient manner (fast wavelet transform, or FWT), as illustrated in the Mallat's herringbone algorithm (Mallat 1989; Gonzalez and Woods 2008), which became the standard DWT method for signal decomposition/expansion and synthesis.

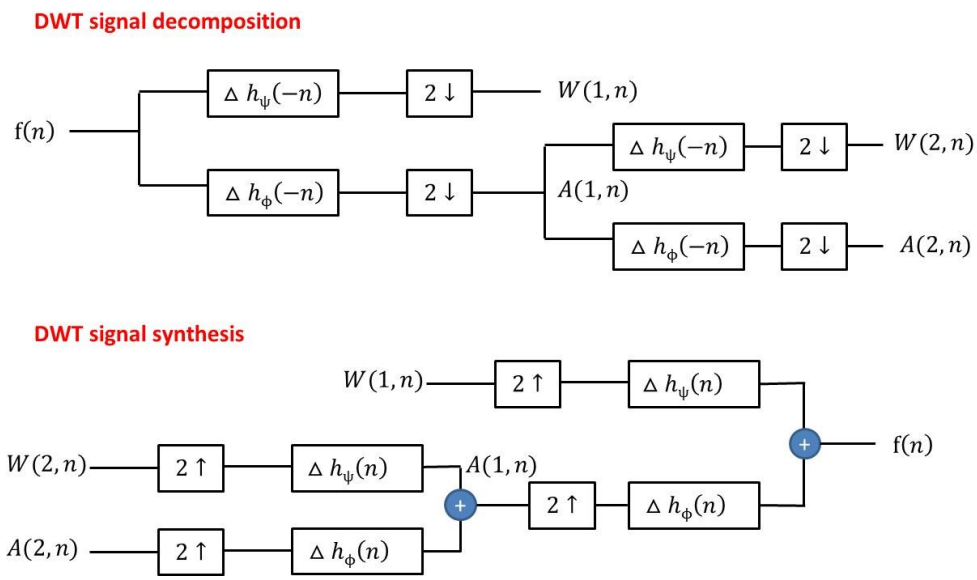


Figure 2-37 : FWT approach to compute the DWT for signal decomposition and synthesis (adapted from Gonzalez and Woods (2008))

The DWT concept can be extended to the case of continuous functions using the continuous wavelet transform (CWT). Practically, for a sampled signal, the CWT involves a convolution point-by-point between the signal and the wavelet filter for all scales, thus producing wavelet planes (W) having the same number of wavelet coefficients than the number of points in the original signal without any upsampling or downsampling. The continuous wavelet transform is defined by the point-by-point convolution of the signal with the wavelet (Gonzalez and Woods 2008):

$$W(s,\tau) = \int_{-\infty}^{\infty} f(x)\, \psi_{s,\tau}(x)\, dx \tag{2-37}$$

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\, \psi\left(\frac{x-\tau}{s}\right) \tag{2-38}$$

In which s and τ are the scale and translation parameters. For a finite signal, the translation parameter will vary so the wavelet is convolved over the full length of the signal (point-by-point). An example of the wavelet decomposition of a signal ($f(x)=\cos(x/(4\pi))$) is shown in Figure 2-38 using the DWT and the CWT (using the Wavelet Toolbox® in Matlab®). The Daubechies 1 (Db 1) wavelet was used to perform a DWT over 5 scales and a CWT over 64 scales.

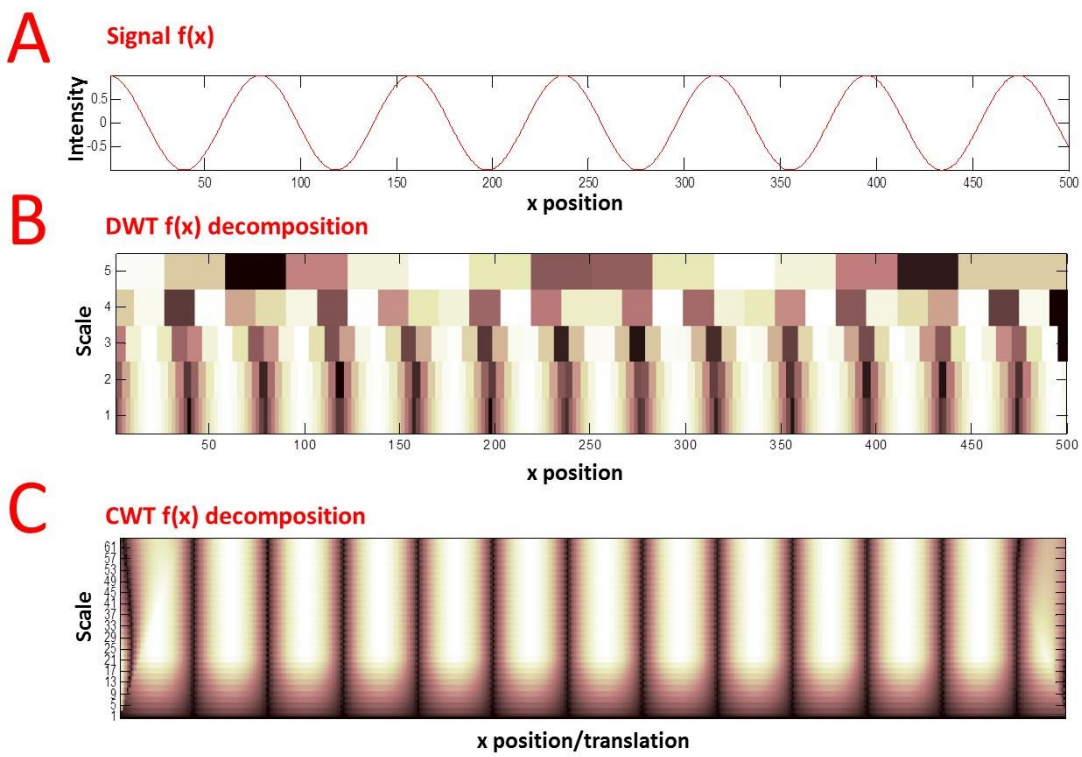


Figure 2-38 : Sinusoidal 1D signal (A) submitted to a DWT (B) or a CWT(C) decomposition. Detail coefficients for both decompositions are shown using a color map: strong positive values are in white and strong negative values are in black.

Figure 2-38 shows that the CWT has a better resolution in terms of spatial position/scale than the DWT, even though it is more complex to compute. In fact, the advantage of the CWT over the DWT is that it has a constant resolution over all scales (even at high scales), since it avoids dyadic subsampling. Also, it is possible to note that the number of detail coefficients stays the same for all scales with the CWT (same as the number of

points in the original signal). In comparison, the number of detail coefficients decreases with the DWT as the scale increases (due to dyadic subsampling).

2.4.6.2 2D - DWT

The DWT transform can be extended to a 2D signal (image) using the same principles (2D-DWT). The main difference being that in the 2D case, one may now consider extracting orientation information in addition to spectral information. For an image of MxN pixels the expansion can be expressed up to scale *j* using the following equation (Gonzalez and Woods 2008):

$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_{m} \sum_{n} A(j_0, m, n) \varphi_{j_0,m,n}(x,y) + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_{m} \sum_{n} W^i(j,m,n) \psi^i{}_{j,m,n}(x,y)$$

(2-39)

To take into account the orientation of the signal, 3 directions are used (Horizontal(H), Vertical(V) and Diagonal(H)), with the 2D scaling and wavelet functions obtained by the product of their 1D counterparts (Gonzalez and Woods 2008):

$$\varphi(x,y) = \varphi(x)\varphi(y) \tag{2-40}$$

$$\psi^H(x,y) = \psi(x)\varphi(y) \tag{2-41}$$

$$\psi^V(x,y) = \varphi(x)\psi(y) \tag{2-42}$$

$$\psi^D(x,y) = \psi(x)\psi(y) \tag{2-43}$$

In the 2D-DWT, A(j,m,n) are approximation coefficients and W(j,m,n) are detail/wavelet coefficients computed by convolution in the spatial domain between the filters and the original image f(x,y) (Gonzalez and Woods 2008):

$$A(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{j_0,m,n}(x,y)$$

(2-44)

$$W^i(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi^i{}_{j,m,n}(x,y) \qquad i = \{H, V, D\}$$

(2-45)

The scaling and wavelet functions must again be scalable to scale *j*, as illustrated in the following equations (Gonzalez and Woods 2008):

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi(2^j x - m, 2^j y - n) \tag{2-46}$$

$$\psi^i{}_{j,m,n}(x,y) = 2^{j/2}\psi^i\left(2^j x - m, 2^j y - n\right) \qquad i = \{H, V, D\} \tag{2-47}$$

Since the scaling and wavelet functions are basically the same as for the 1D-DWT and since they also form an orthonormal basis, the same scaling vector $h_\varphi(n)$ and wavelet vector $h_\psi(n)$ can be used for the 2D-FWT in a multistage structure (considering this time the orientation of the signal), relying on the strategy proposed by Mallat (1989). The FWT/DWT for 1D and 2D signals was first introduced by Mallat (1989), for which the computation of the wavelet representation is accomplished with a pyramidal algorithm based on convolutions with quadrature mirror filters. The DWT is in fact a series of high and low pass filtering operations, between which a subsampling of a factor of 2 is applied on the approximation image before going to the next level (Caponetti, Castellano et al. 2009). The DWT is optimal for image compression (the downsampling reduces the size of the wavelet coefficient matrices to be saved), but not optimal for filtering, deconvolution, detection or pattern recognition (Starck, Elad et al. 2004). For 2D images, this method is illustrated in Figure 2-39 for signal decomposition and synthesis.

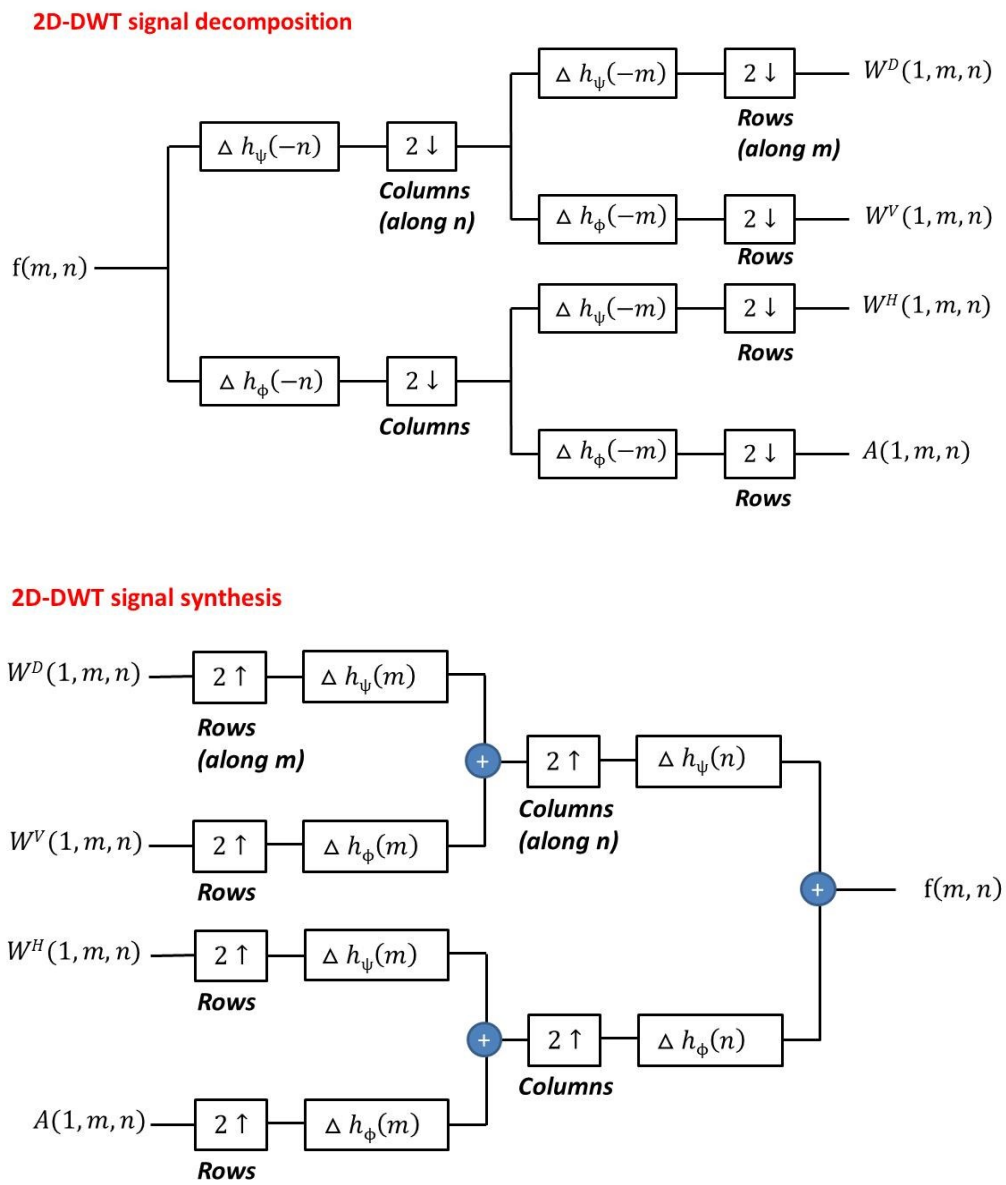


Figure 2-39 : FWT approach to compute the DWT for 2D signal decomposition and synthesis (adapted from Gonzalez and Woods (2008))

Using the Wavelet Toolbox® in Matlab®, a typical cell image was analyzed to generate a series of wavelet planes (W), as illustrated in Figure 2-40.

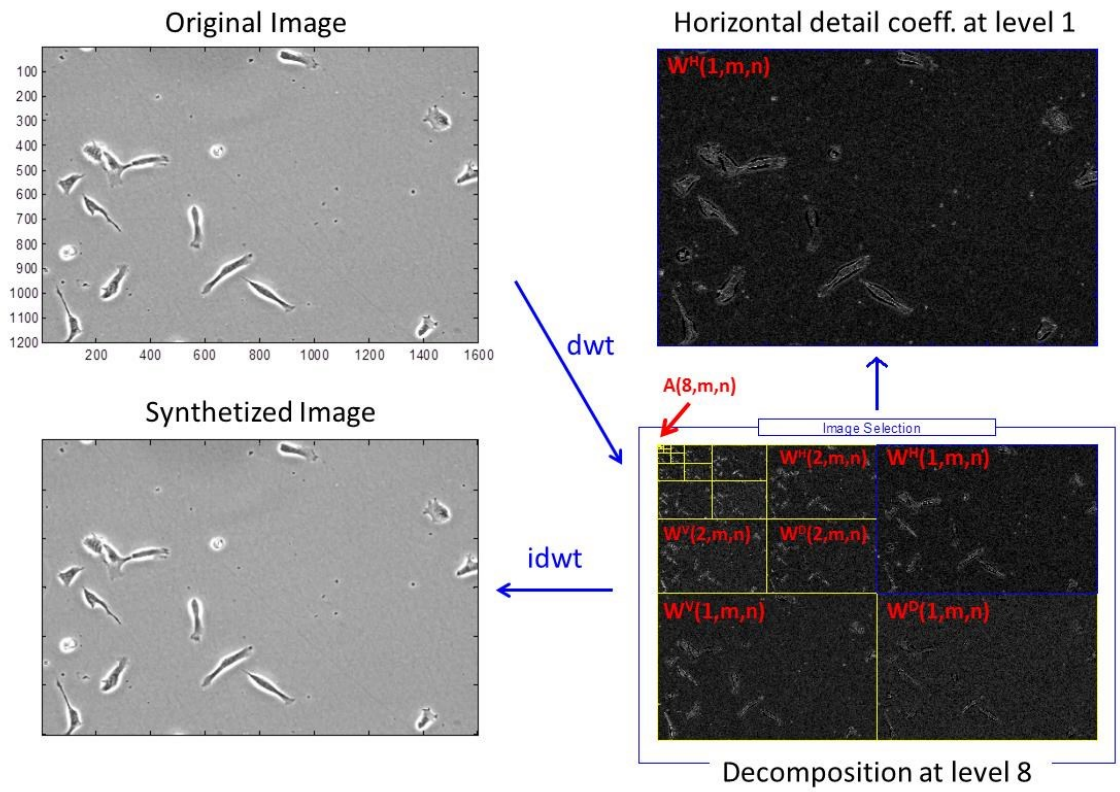


Figure 2-40 : 2D-DWT applied to a typical cell image

As shown in Figure 2-40, the 2D-DWT generates wavelet detail subimages (W, equivalent to wavelet planes for the CWT) that are of different sizes because of the downsampling operation between scales. Since the different wavelet planes from the DWT are not congruent, this makes it difficult to localize the textural information over several scales (Bharati, Liu et al. 2004). While many wavelet applications in image processing use Mallat's orthogonal wavelet transform (DWT) (Mallat 1989), this transform is not always optimal for pattern recognition because it is not shift-invariant (due to decimation) and because image subbands (captured by each wavelet plane) are uncorrelated across scales (Olivo 1996; Olivo-Marin 2002). To avoid the drawbacks associated to the DWT, it is possible to use instead the undecimated wavelet transform (UWT), which is more convenient for pattern recognition since it is translation invariant and since the wavelet coefficients in each wavelet plane (W) are correlated and redundant (same number of pixels than in the original image). The UWT is very simple to implement for signal decomposition and synthesis (Olivo 1996).

2.4.6.3 2D – UWT

To compute approximation and detail images from the UWT, it is possible to use the *à trous algorithm* (Starck and Bijaoui 1994; Olivo 1996) associated with the stationnary wavelet transform (SWT), in which the scaling and wavelet vectors are scaled-up by a factor of 2 at each scale. The SWT also allows extracting orientation features (vertical, horizontal and diagonal wavelet coefficients). However, this approach is still limited in scale/spatial resolution because of the filters upsampling steps between each scale. To avoid this problem and to perform the convolution at the single pixel level (point by point), the approximation images can be computed directly in the frequency domain (UWT-CWT), using the FFT transform of the scaling function. The Fourier transform of the low pass filter associated with the Gaussian wavelet (scaling function) is computed using the codes available from the YAWTb website (Jacques, Coron et al. 2007). This filter is based on the following equation for each scale *s* and for standard deviation *σ* (Jacques, Coron et al. 2007):

$$\Phi_{\mathrm{Gaussian}}(\mathrm{u},\mathrm{v},\sigma,\mathrm{s}) = \mathrm{e}^{-\sigma^2((\mathrm{su})^2+(\mathrm{sv})^2)/2} \tag{2-48}$$

Note that *u* and *v* are expressed in rads/pixel. The approximation coefficients are computed by multiplying point-by-point the Fourier transform of the image (F(u,v)) with the Fourier transform of the scaling functions (equivalent to a convolution in the Euclidean space (Gonzalez and Woods 2008)), and by transforming the result back into the Euclidean domain (FFT^{-1}, standing for the inverse Fourier transform). It is thus possible to compute the Gaussian approximation image after convolution with a low pass filter ($A_{Gaussian}$):

$$\mathrm{A}_{\mathrm{Gaussian}}(\mathrm{x},\mathrm{y},\sigma,\mathrm{s}) = \mathrm{FFT}^{-1}[\mathrm{F}(\mathrm{u},\mathrm{v}).*\,\Phi_{\mathrm{Gaussian}}(\mathrm{u},\mathrm{v},\sigma,\mathrm{s})] \tag{2-49}$$

Each Gaussian wavelet plane can be computed by taking the difference between two consecutive approximations (Olivo 1996), thus capturing the information located in a precise frequency bandwith. The $W_{Gaussian}$ can thus be seen as a Difference-of-Gaussian (DoG) wavelet plane:

$$\mathrm{W}_{\mathrm{Gaussian}}(\mathrm{x},\mathrm{y},\sigma,\mathrm{s}) = \mathrm{A}_{\mathrm{Gaussian}}(\mathrm{x},\mathrm{y},\sigma,\mathrm{s}) - \mathrm{A}_{\mathrm{Gaussian}}(\mathrm{x},\mathrm{y},\sigma,\mathrm{s}+1) \tag{2-50}$$

Signal synthesis can be performed by summing all the Gaussian wavelet planes (up to s_{max}) and combining the result with the residual approximation at s_{max} (Olivo 1996):

$$f(x,y) = A_{Gaussian}(x,y,\sigma,s_{max}) + \sum_{s=1}^{s_{max}} W_{Gaussian}(x,y,\sigma,s)$$

(2-51)

Alternatively, one can choose the two-dimensional Gabor scaling function which consists of a sinusoidal wave of a certain frequency and orientation modulated by a Gaussian envelope (Tuceryan and Jain 1999). The Fourier transform of the Gabor wavelet (scaling function) is also computed based on codes available from the YAWTb website (Jacques, Coron et al. 2007). The Gabor wavelet is mathematically expressed by the following equations for each scale *s* and angle *θ* and for *σ*, k_0 and *ε* (Jacques, Coron et al. 2007):

$$\psi_{Gabor}(u,v,\sigma,\varepsilon,k_0,s,\theta) = e^{-\sigma^2((u_r-k_0)^2+(\varepsilon v_r)^2)/2} \quad (2\text{-}52)$$

$$u_r = s * \left(u * \cos(\theta) - v * \sin(\theta)\right) \quad (2\text{-}53)$$

$$v_r = s * \left(u * \sin(\theta) + v * \cos(\theta)\right) \quad (2\text{-}54)$$

As *σ* (width of the Gaussian envelope) influences the sharpness of the wavelet, k_0 (offset from the zero frequency) impacts the frequency within the wavelet and *ε* determines the spread of the wavelet in the spatial domain. Figure 2-41 shows the impact of these parameters on a typical Gabor wavelet.

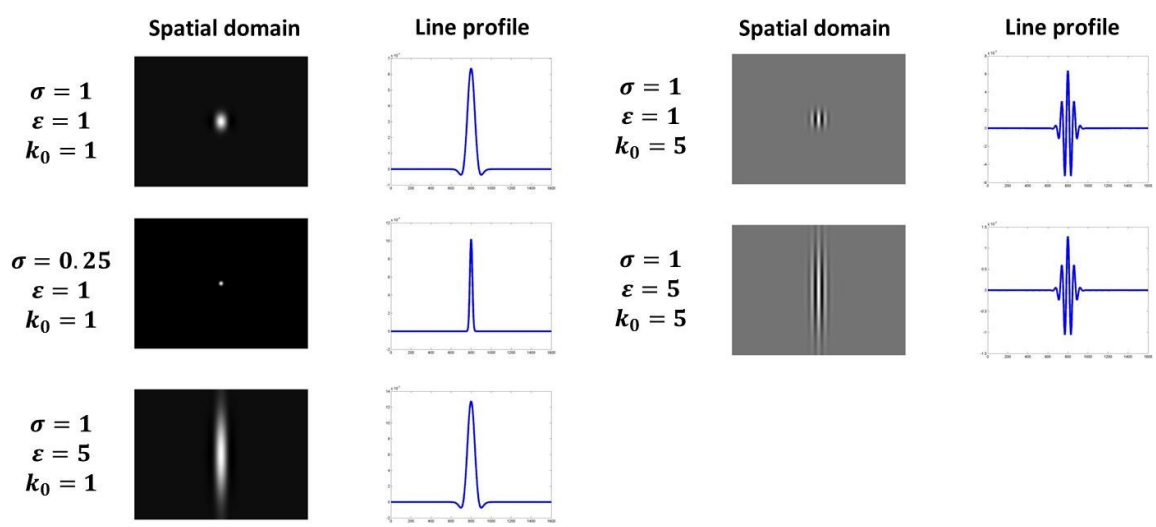


Figure 2-41 : Impact of σ, ε and k_0 on the shape of a typical Gabor wavelet (s = 50, θ = 0)

The Gabor wavelet is a bandbass filter located in a specific area of the Fourier domain. Wavelet planes obtained by applying the Gabor wavelet are computed as follows (Jacques, Coron et al. 2007), capturing the information in the Fourier domain at a specific angle for a specific bandwith:

$$W_{Gabor}(u, v, \sigma, \varepsilon, k_0, s, \theta) = FFT^{-1}[F(u, v).* \psi_{Gabor}(u, v, \sigma, \varepsilon, k_0, s, \theta)] \quad (2\text{-}55)$$

The UWT-CWT decomposition based on the Gabor wavelet can be used to generate a series of wavelet planes having the same size than the original image (i.e. spatially congruent images), capturing textural details over several scales and for different orientations (from specific regions in the FFT transform of the image).

The UWT-CWT decomposition is illustrated in Figure 2-42 for the Gaussian wavelet (σ=0.5) and in Figure 2-43 for the Gabor wavelet (σ=0.5, ε=5, k_0=1, θ=π/4) using a typical cellular image. It is possible to see that the Gaussian wavelet can be used to detect and assess the size of round objects that are isotropic. In comparison, the Gabor wavelet can be used to detect and assess the size of elliptical and anisotropic objects, capturing also orientation information. Since adherent cells (such as myoblasts) are quite anisotropic, this explains why the Gabor wavelet was preferred for this PhD project.

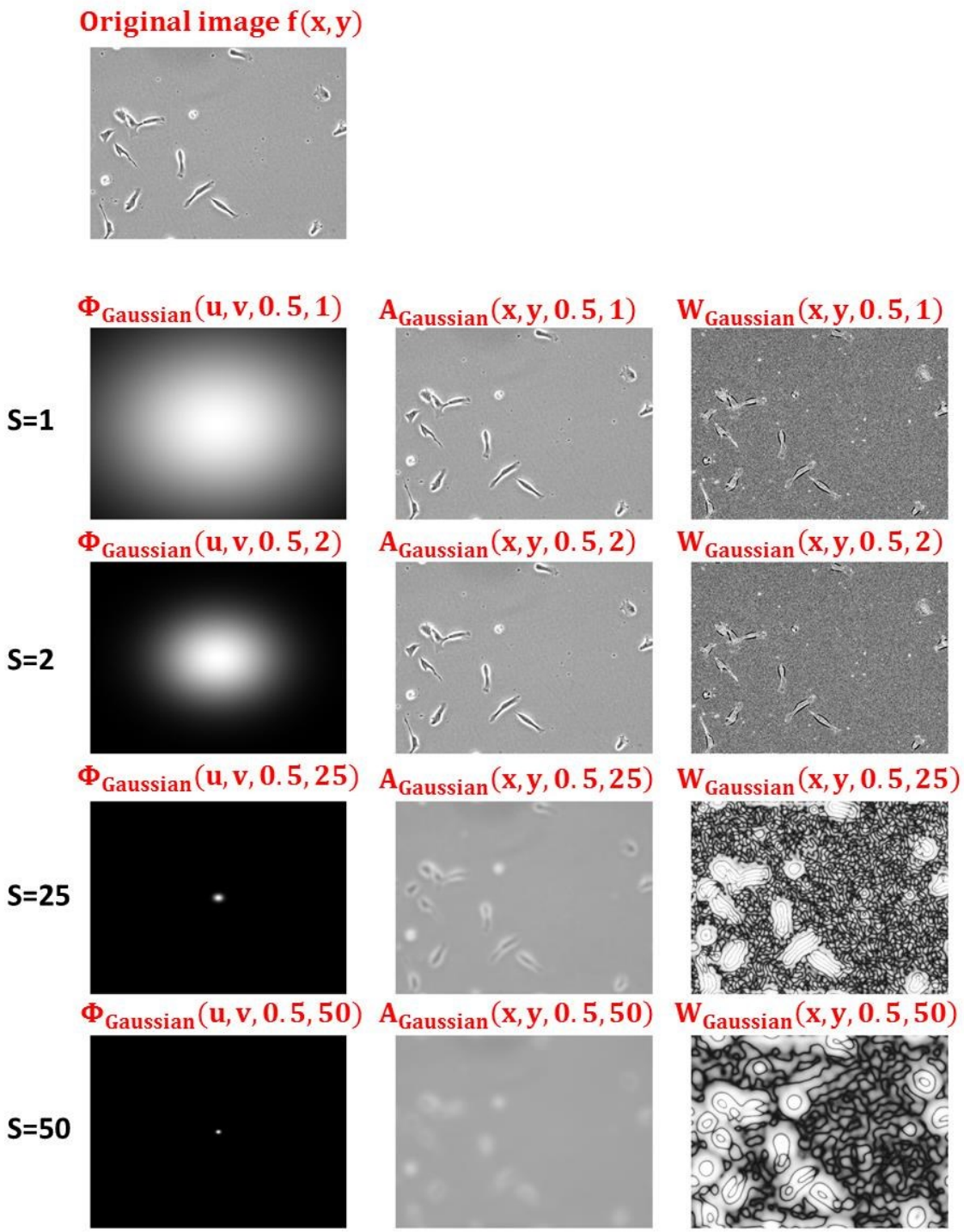


Figure 2-42 : UWT-CWT decomposition using the Gaussian wavelet on a typical cell image

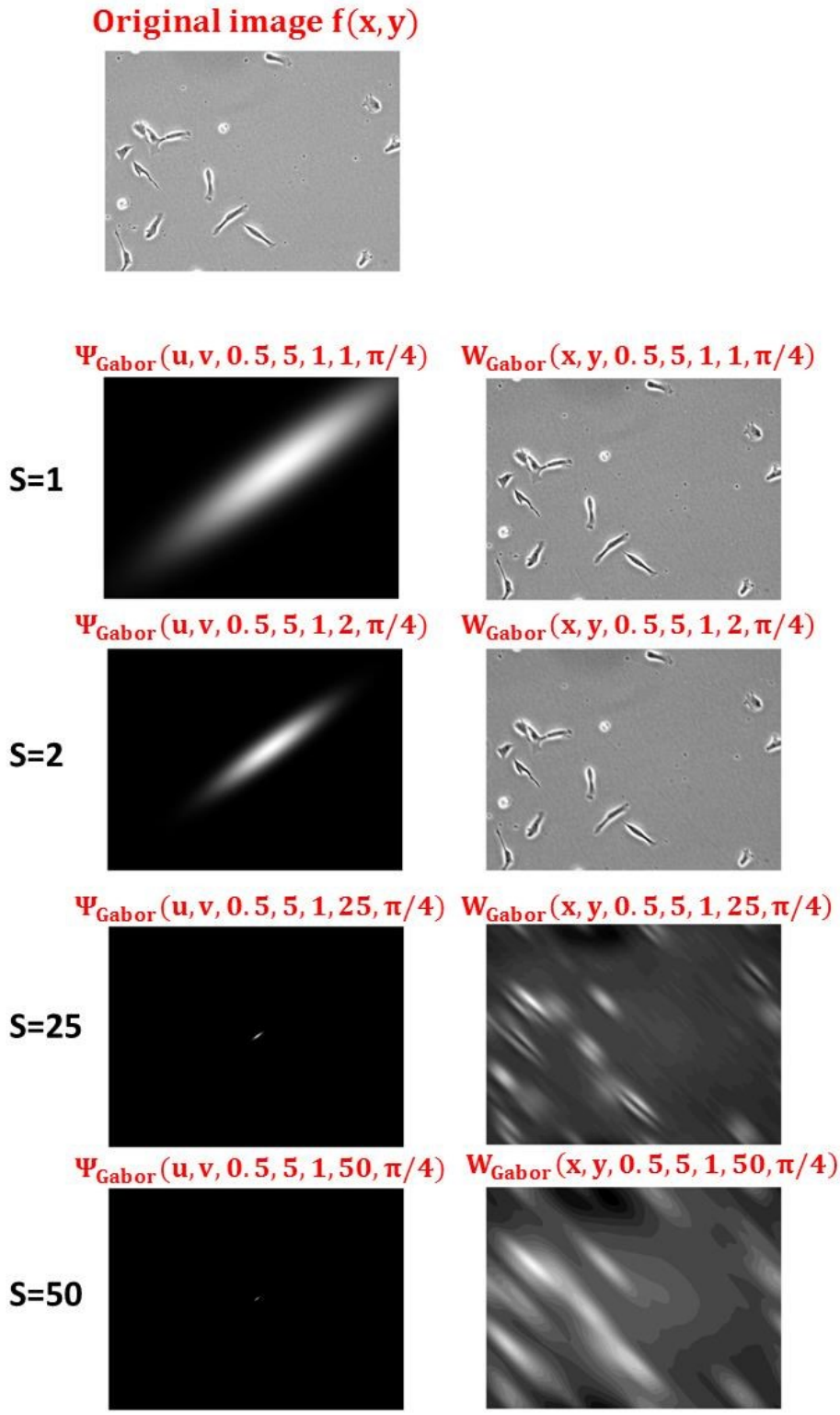


Figure 2-43 : UWT-CWT decomposition using the Gabor wavelet on a typical cell image

Since the Gaussian and the Gabor wavelet planes are computed from a point-by-point convolution in the frequency domain, these UWT-CWT transforms have the potential of providing better spatial/frequency resolution compared to the 2D-DWT, while providing the opportunity to look at the signal orientation more precisely (θ can be adjusted as wanted).

The UWT-CWT wavelet approach is linked to another multiscale analysis method which is called the scale-invariant feature transform (SIFT) (Lowe 1999; Lowe 2004), for which the image is convolved with Gaussian kernels of different scales and standard deviations σ. The differences between consecutive Gaussian-blurred images are computed as Differences-of-Gaussians (DoG) (Jiang, Crookes et al. 2010). Key points are identified as local extrema in each DoG image, and since they are cell specific, they can be used for cell tracking (Jiang, Crookes et al. 2010). The SIFT features were used to track different cell lines in DIC microscopy images (Jiang, Crookes et al. 2010). SIFT features were also used to detect mitosis in PCM images (Liu, Li et al. 2011). In another approach called the flat texture transform, the difference between the image and a median filtered version of itself is used to detect pixel groups, the size of the filter being an important parameter (Rodenacker and Bengtsson 2003). However, the UWT-CWT has the advantage (compared to the SIFT and to the flat texture transform) of being fast (convolution computed in the Fourier domain) while looking at the direction of the signal (with the Gabor wavelet) over several scales. Finally, if the images are highly anisotropic and if orientation is important, other transforms such as ridgelets, curvelets, bandlets and contourlets transforms (other variations of the wavelet transform) could be used, involving the analysis of approximation images generated by the UWT and the DWT (Starck, Candes et al. 2002; Starck, Elad et al. 2004). Ridgelets and curvelets were developed to compensate the weakness of separable wavelets to represent lines and curves in 2D signals (Starck, Elad et al. 2004).

2.4.6.4 Gaussian wavelet cutoff frequency (F_0)

For a low pass filter expressed in the frequency domain, the point of transition between the high frequencies that are cut and the low frequencies that are kept is called the cutoff frequency (F_0) (Gonzalez and Woods 2008). The Gaussian low pass filter can be expressed as a function of F_0 as follows (Gonzalez and Woods 2008):

$$\Phi_{\mathrm{Gaussian}}(u, v, D_0) = e^{-(u^2+v^2)/2{F_0}^2} \qquad (2\text{-}56)$$

When comparing this expression to the Gaussian wavelet scaling function (equation (2-48)), it is possible to deduce that:

$$-\frac{\sigma^2 s^2}{2} = -\frac{1}{2{F_0}^2} \tag{2-57}$$

The cutoff frequency (F_0) can then be expressed by the following equation (in rads/pixel):

$$F_0 = \pm\frac{1}{\sigma s} \tag{2-58}$$

The cutoff frequency (F_0) can also be expressed in cycles/pixel as following:

$$F_0 = \pm\frac{1}{2\pi\sigma s} \tag{2-59}$$

The signal period associated to the cutoff frequency (P_0, in pixels) can be calculated by inverting the precedent equation:

$$P_0 = 2\pi\sigma s \tag{2-60}$$

This relation establishes a link between the scale *s*, the standard deviation *σ* of the Gaussian wavelet used and the size of the details captured in pixels. For example, for a Gaussian wavelet with *σ* = 0.5, the periods associated to the signal captured at each scale is given in Table 2-6.

Table 2-6 : Gaussian wavelet cutoff frequencies and periods (for σ = 0.5)

Scale s	F_0 (cycles/pixel)	P_0 (pixels)
0	Infinity	0
1	0,32	3,14
2	0,16	6,28
3	0,11	9,42
4	0,08	12,57
5	0,06	15,71
6	0,05	18,85
7	0,05	21,99
8	0,04	25,13
9	0,04	28,27
10	0,03	31,42
11	0,03	34,56
12	0,03	37,70
13	0,02	40,84
14	0,02	43,98
15	0,02	47,12
…	…	…

Resolution limits in the spatial and in the frequency domains are determined by the *Heisenberg* uncertainty inequality (Tuceryan and Jain 1999), where *u* and *v* are expressed in rads/pixel:

$$\Delta x \Delta u \geq \frac{1}{4\pi} \tag{2-61}$$

$$\Delta y \Delta v \geq \frac{1}{4\pi} \tag{2-62}$$

These inequalities have as a consequence that the frequency resolutions increase (Δu and Δv decrease) as the wavelet scale increases while spatial resolutions decrease (Δx and Δy increase). The relationship between frequency resolution, space resolution and wavelet scale is illustrated in Figure 2-44.

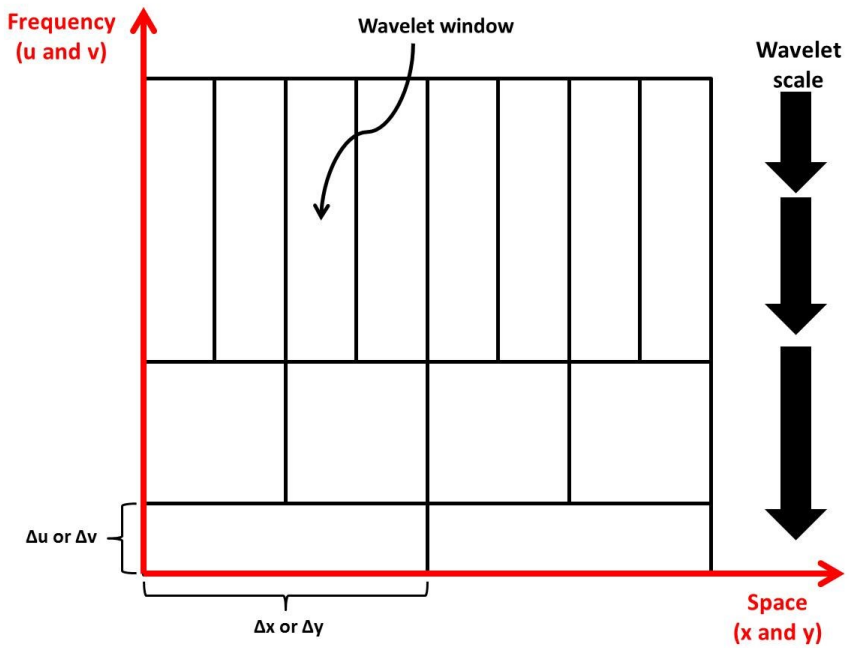


Figure 2-44 : Relationship between frequency resolution, space resolution and wavelet scale

Figure 2-44 also shows that the spatial width of the wavelet window increases as its frequency width decreases as the wavelet gets wider at higher scales (Tuceryan and Jain 1999).

2.4.6.5 Use of wavelets in the literature

The number of articles in the literature reporting the use of wavelets for analyzing microscopic images is very limited. A few examples of the use of wavelets in biological applications (mainly for textural features) are provided in this section. For example, wavelet texture analysis was used to assess the quality of oocytes cells for assisted insemination, which is related to the granularity and the texture of their cytoplasm (Caponetti, Castellano et al. 2009): the classification was performed using fuzzy c-means clustering (Caponetti, Castellano et al. 2009). Textural features were fed to a SVM classifier using a wavelet kernel to classify regions as cell edge, cell interior and background in order to segment V79 cells (Skoczylas, Rakowski et al. 2011). Textural analysis (co-occurrence matrix, Tamura, wavelets, neighborhood difference matrix and GLRLM features) was used in the diagnosis of diffuse gliomas from colored microscopic images (Jun, Cooper et al. 2010) by reducing the features space using a 3 components linear discriminant analysis or LDA (Jun, Cooper et al. 2010). Gabor filters were also used

to segment neuromeres by looking at their texture (Wu, Schulte et al. 2010). The UWT was used to detect and track endocytosed quantum dots in fluorescence images taken in time-lapse confocal microscopy (Genovesio, Liedl et al. 2006). Finally, in signal processing applications, wavelets were used to denoise and segment Hoeschst-stained cell clusters in epifluorescence images (Padfield, Rittscher et al. 2008; Padfield, Rittscher et al. 2011) and were used to help finding appropriate thresholds to segment the edge of marrow cells (Hou, Ma et al. 2007). From these very few examples, it is possible to conclude that the full potential of wavelets for analyzing microscopic images might not be exploited yet.

2.4.7 Classifiers

In biological applications, once image features are computed (regardless of the method used to extract them), they are typically submitted to a pattern recognition algorithm (classification or modelling) in order to perform segmentation (e.g. identify cells) or to differentiate in an automated manner different culture conditions. A pattern recognition approach involves assigning with minimal human intervention measured feature values to their respective pattern classes, which represent sets of typical occurrences that share common properties (Gonzalez and Woods 2008). For example, in this thesis (Chapter 6), shape-based features (cellular morphology) were used to differentiate cells growing in culture medium A from cells growing in culture medium B. There are mainly two types of classifiers available to perform pattern recognition: unsupervised classifiers and supervised classifiers. A review of both types of classifiers and pattern recognition methods was presented by Jain, Duin et al. (2000). Most classification methods rely on a separation performed in the features space, as illustrated in Figure 2-45. A features space can be seen as a set of axes defining an *n*-dimensional Euclidean space, and each feature value can be seen as a coordinate on each axis (one coordinate per feature). Here only three dimensions are shown, since it is difficult to illustrate more than three dimensions.

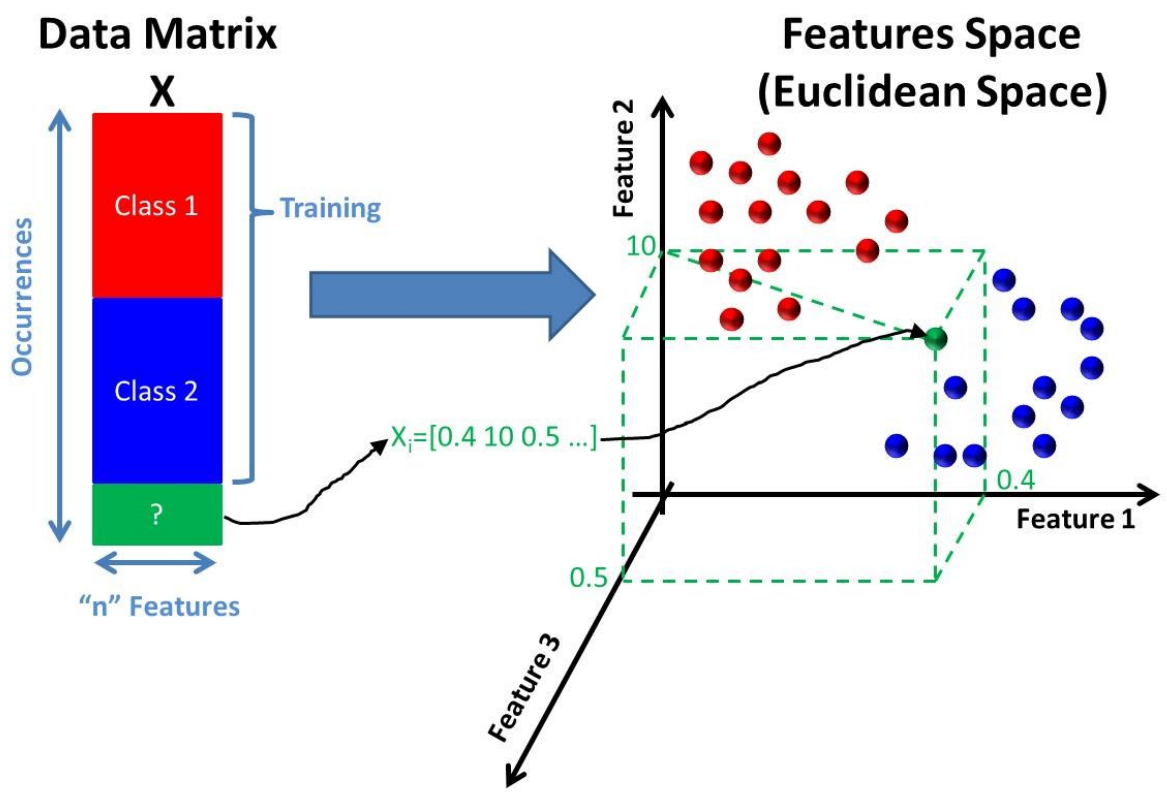


Figure 2-45 : Features space seen as an *n*-dimensional Euclidean space

Of particular interest for this thesis, multivariate data analysis techniques (principal component analysis, partial least squares) will be detailed in Section 2.5, since they are used in pattern recognition to process features from continuous 2D wavelets (using multivariate image analysis or MIA) for classification and modelling purposes.

2.4.8 Parameters calibration

Most image processing algorithms used for cell segmentation and tracking rely on parameters specified by the user or automatically adjusted through a training process to perform their task. These parameters can be a threshold value, the size of a filter, a standard deviation used in a probability distribution for a stochastic model, etc. These parameters can remain constant for all cases (if the algorithm is robust for all cases), they can be set by trial-and-error by the user, they can be automatically adjusted through a recursive algorithm (unsupervised) or they can be adjusted through a calibration/training process (supervised). Even though the calibration of a segmentation model is a critical step, parameter estimation is under-reported in the literature (Al-Muhairi, Fleury et al. 2011). To assess the performance of segmentation algorithms, a literature review was

proposed by Zhang (1996) in which evaluation methods are presented as analytical, based on the goodness of fit or based on empirical discrepancy. Analytical methods study segmentation algorithms by considering their principles, requirements, utilities and complexity, without the need of any implementation or test on real images (Zhang 1996). Goodness of fit methods evaluate the performance of segmentation algorithms based on the quality of the segmented images (goodness based on intra-region uniformity, on inter-region contrast or on region shape) (Zhang 1996). Empirical methods measure the discrepancy between the segmented image (by the algorithm) and a reference (ground truth) image, and the discrepancy can be based on the number of mis-segmented pixels, on the position of the mis-segmented pixels or on the number of objects detected (Zhang 1996).

In this PhD project, since PCM microscopic images can vary in terms of quality (uneven illumination, bright halo surrounding the cells, cells touching, etc.), we opted for an empirical approach to evaluate and tune segmentation algorithms (see Chapter 4). To train a segmentation or a tracking algorithm through an empirical calibration process (to adjust its parameters), it is required to have a good training dataset (ground truth images or sequences) and to use an objective function to assess the performance of the algorithm relative to parameter values.

2.4.8.1 Training dataset

Calibration usually requires a set of calibration images or time-lapse sequences (referred to as “ground truth”). For segmentation algorithms, a ground truth dataset can be generated manually (by human observers) or automatically (synthetic data). Using manually segmented ground truth images is a common way of benchmarking and tuning segmentation algorithms, but can be fastidious and the results may vary between observers who perform the segmentation task. Some benchmark ground truth images are freely available on the internet (Quanli 2010). Several authors use manual ground truth images to tune segmentation algorithms. For example, ground truth images (“learning images”) were used to train a classifier before analyzing a whole PCM image dataset (Skoczylas, Rakowski et al. 2011). If generating manual ground truth images is difficult even for an experimented human observer (like in epifluorescent images with a low signal-to-noise ratio compared to confocal microscopy images), an edit-based validation approach can be used (Padfield, Rittscher et al. 2008). When testing a segmentation

model for brightfield or PCM images, one can use fluorescent stained images as ground truth (Selinummi, Ruusuvuori et al. 2009).

When using more than one human observer/user to segment or classify cells, some variability in the ground truth results should be expected. For example, when classifying cells imaged in PCM according to their morphology, Theriault, Walker et al. (2012) asked a first observer to look at the cells, and found 82.8% spread cells, 60.5% polarized cells and 31.9% oriented cells. A second observer found 84.0% spread cells, 63.9% polarized cells and 17.0% oriented cells, meaning that even though the two observers labeled the same 838 cells, they completely agreed on their morphology on less than 60% of them (Theriault, Walker et al. 2012).

To be more efficient and to avoid inter-observer variability, several authors use synthetic ground truth images by incorporating objects (cells) that are similar to the real objects to segment. Usenik, Vrtovec et al. (2011) generated an artificial ground truth image (usually sinusoidal functions for cells), and tested their algorithm on it, looking at the influence of noise on the results. Synthetic ground truth images of neuritis were used to test the performance of an image analysis algorithm (Wu, Schulte et al. 2010). Synthetic ground truth images were also used to test a segmentation and denoising algorithm used to segment bright spots for immunochemistry assays (Olivo-Marin 2002), and to test a new watershed-based segmentation algorithm (Cheng and Rajapakse 2009).

To characterize the efficiency and accuracy of a segmentation algorithm in PCM, synthetic ground truth images of different signal-to-noise ratios (SNR) and using different cell shapes were generated (Ambuhl, Brepsant et al. 2012). They were generated by simulating the optics of PCM (Yin, Li et al. 2010) using the parameters provided by the microscope manufacturer and applying an image formation model to the segmentation maps of different imaged cells serving as ground truth objects (Ambuhl, Brepsant et al. 2012). A total of five different shapes were used, incorporating the degree of polarization of the cells (Ambuhl, Brepsant et al. 2012).

2.4.8.2 Performance metrics

A theoretical overview of the different metrics available for assessing the performance of segmentation algorithms and classifiers (i.e. binary decision problems) was proposed by Hand et al. (2012). The literature relative to this subject is very diverse, and the metrics used are application or problem specific (Hand 2012). Often there seems to be little awareness that the same classification problems arise in other branches of science and

technology, having as a consequence that the same performance measures are duplicated under different names (Hand 2012).

2.4.8.2.1 Confusion matrix metrics for binary decision problems

In a binary decision problem, a classifier labels occurences as either positive or negative, and this decision can be represented in a structure called *confusion matrix* or *contingency table* (Davis and Goadrich 2006). In the confusion matrix, true positives (TP) are occurences correctly labeled as positives, false positives (FP) are occurences incorrectly labeled as positives, true negatives (TN) are occurences correctly labeled as negative and false negative (FN) are occurences incorrectly labeled as negative (Davis and Goadrich 2006). For a segmentation algorithm trying to distinguish cellular regions from background regions, a typical confusion matrix is provided in Figure 2-46. In this example, a ground truth image segmentation map (considered as the true segmentation map) is compared to the segmentation algorithm segmentation map to calculate the number of TP, FP, TN and FN (in pixels).

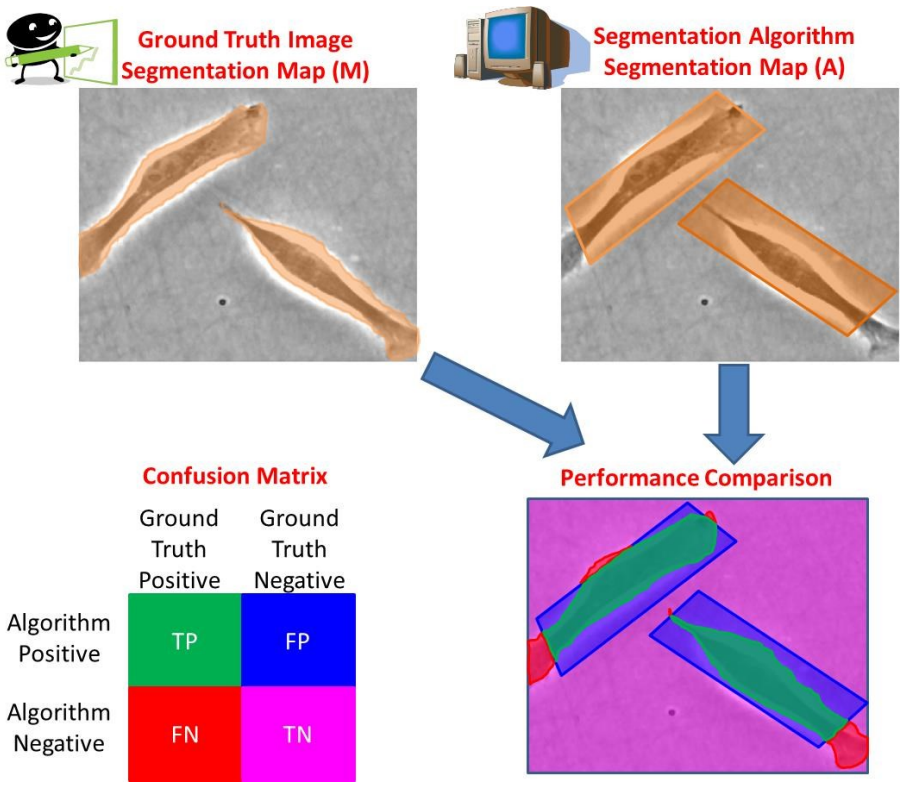


Figure 2-46 : Typical confusion matrix for a segmentation algorithm used to differentiate cellular regions from background regions

Once the numbers of TP, FP, FN and TN are computed, one can compute several ratios of interest providing an idea of how well an algorithm is performing. Several ratios can be used to quantify the segmentation performance: sensitivity (Se), recall (R), positive predictive value (PPV), precision (P), negative predictive value (NPV), true positive rate (TPR), false positive rate (FPR), specificity (Sp) and classification accuracy (ACC). The following equations show how to compute these ratios (Davis and Goadrich 2006; Burgemeister, Nattkemper et al. 2010):

$$\mathrm{Se} = \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{2-63}$$

$$\mathrm{PPV} = \mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{2-64}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{FN+TN}} \tag{2-65}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{2-66}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \tag{2-67}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{FP+TN}} \tag{2-68}$$

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FN+TN+FP}} \tag{2-69}$$

From a matrix perspective, considering the binary segmentation map obtained by manual segmentation (M) and the binary segmentation map obtained by automatic segmentation (A), recall (R) and precision (P) are defined as follows (He, Wang et al. 2007):

$$\mathrm{R} = \frac{\sum\sum(\mathrm{A}.*\mathrm{M})}{\sum\sum(\mathrm{M})} \tag{2-70}$$

$$\mathrm{P} = \frac{\sum\sum(\mathrm{A}.*\mathrm{M})}{\sum\sum(\mathrm{A})} \tag{2-71}$$

In which ".*" involves a point-by-point multiplication between two matrices. Since several of these ratios (such as R and P, Sp and Se, etc.) evolve in opposite directions, it is required to use a weighted objective function (classification accuracy criteria) that can reach a compromise between them (Davis and Goadrich 2006). For example, the Se and the Sp ratios of an algorithm which performs nuclear detection in 4D Nomarski DIC microscopy evolve in opposite directions and depend on the parameters used (namely the window size

of the entropy filter and the entropy threshold): the higher the Se, the lower the Sp (Hamahashi, Onami et al. 2005). The adjustment of the segmentation algorithm parameters can thus be seen as an optimization problem.

2.4.8.2.2 Classification accuracy criteria for binary decision problems

Since several ratios described earlier evolve in opposite directions (increasing one will decrease the other), some compromise has to be reached so that both classes (TP and TN) are taken into account (Hand 2012). Evaluating the performance of an algorithm can be an issue if one of the classes is very small compared to the other ("unbalanced classes" (Hand 2012)). To tackle this problem, it is possible to combine the Se and the PPV or the R and P and to compute their harmonic mean, called the F-measure or F_M (Hand 2012):

$$F_M = 2(Se^{-1} + PPV^{-1}) = \frac{2*Se*PPV}{Se+PPV} \quad (2\text{-}72)$$

$$F_M = 2(R^{-1} + P^{-1}) = \frac{2*R*P}{R+P} \quad (2\text{-}73)$$

The F-measure function can sometimes be averaged relative to reverse classification in order to compute the mean F-measure (F_s) (Hand 2012):

$$F_S = \frac{Se*PPV}{Se+PPV} + \frac{Sp*NPV}{Sp+NPV} \quad (2\text{-}74)$$

Some methods use an objective function such as a Berkeley F-function (F) for boundary pixels (Al-Muhairi, Fleury et al. 2011), which is a weighted function between P and R. This function (to be maximized) is expressed by:

$$F = \frac{P*R}{\alpha*R+(1-\alpha)*P} \quad (2\text{-}75)$$

Most of the time an α value of 0.5 is used (Selinummi, Ruusuvuori et al. 2009). However the F-measure or F-function does not have a straightforward probability interpretation (Hand 2012).

2.4.8.2.3 Classification accuracy curves for binary decision problems

Metrics described so far are computed from classification results obtained for a specific set of parameters for a specific classifier or segmentation algorithm. To compare algorithms, it is good pratice to test them on ground truth data obtained in different experimental conditions to see how the different ratios evolve. Receiver Operator Characteristic (ROC)

and Precision-Recall (PR) curves are typically generated to evaluate the performance of one or several machine learning algorithms on a given dataset (Davis and Goadrich 2006). The relationship between PR and ROC curves is studied in Davis and Goadrich (2006).

ROC curves are generated by varying one parameter at a time for each algorithm (usually a classifier threshold) over a specified range or distribution (Hand 2012), and by looking at the classification performance for each parameter value. The ROC curve (usually Se against 1-Sp) shows how the number of correctly classified positive occurences varies with the number of incorrectly classified negative occurences (Davis and Goadrich 2006). A typical ROC curve comparing two algorithms is shown in Figure 2-47:

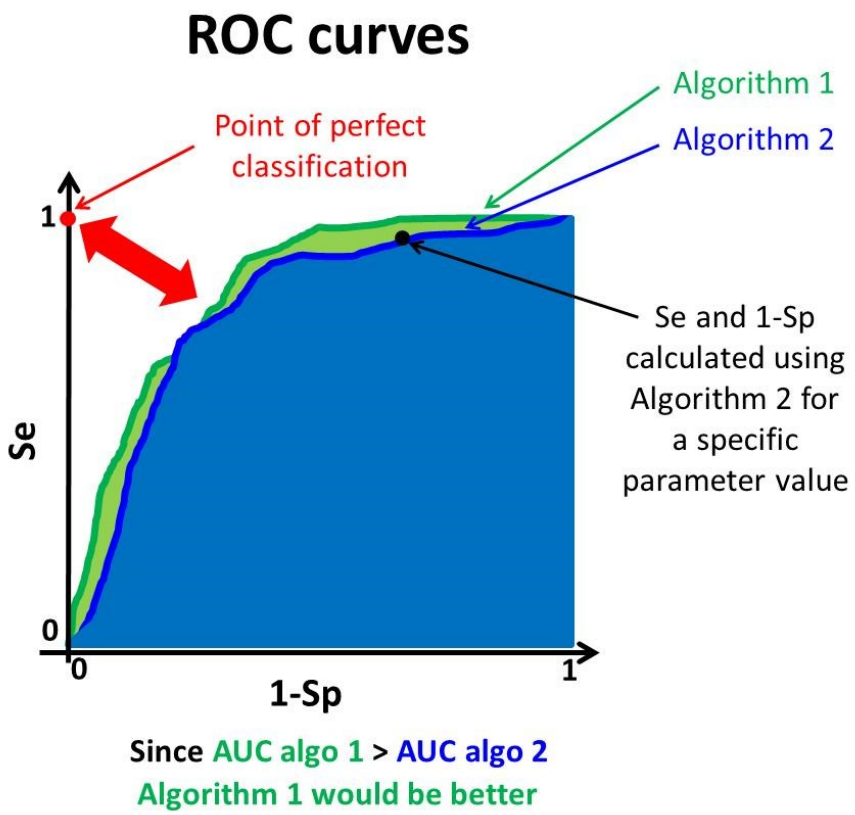


Figure 2-47 : Example of ROC curves for comparing two algorithms for classification or segmentation

Each point on each curve represents a classification performance achieved for a specific parameter value. Basically, the closer the curve is from the point of perfect classification, the better the algorithm is. Often the area under the curve (AUC, named the *c-statistic* in medicine) is used as a simple metric to compare algorithms performing binary decisions

(Hand 2012), and is simply the area under the ROC curve. The latter can be computed using trapezoidal areas created between the curve points (Hand 2012).

Other metrics (more problem-specific) have also been proposed. For example, for several segmentation tasks, the delimitation of the boundary of the objects/cells/tissues is critical, and distance-based metrics can be used to measure the distance between the segmentation map boundary generated by the algorithm and the ground truth segmentation map boundary (Fenster and Chiu 2005).

2.4.8.2.4 Performance metrics for complex problems

When analyzing images or time-lapse sequences of images, more complex algorithms can be used, performing decisions or classifications that are not necessarily binary. Also, complex models can be built by tuning an algorithm that estimates a set of variables of interest (**Y**) from a set of predictors (**X**). For more complex models, other strategies must be used to assess their performance and to tune them, and the literature on model/algorithm selection is quite large. To compare algorithms and models (sets of parameters), the mean square error (MSE) on predicted variables (**Y**) or the Akaike Information Criterion (AIC) can be used (Akaike 1974; Tabatabai, Bursac et al. 2011).

To assess the robustness and performance, the dataset can be split to be analyzed into two groups: a training dataset (used to tune the parameters of the algorithm) and a validation dataset (to see if the algorithm offers a good prediction performance). Several multivariate and model identification methods available in the literature can be used to assess the performance of an algorithm with a specific set of parameter values, such as cross-validation, jackknifing and bootstraping, involving different sampling or resampling strategies (Duchesne and MacGregor 2001). To evaluate the performance of classifiers used for segmentation, cross-validation (Jun, Cooper et al. 2010) and jackknifing (Malpica, Santos et al. 2003) can be used. For example, to train classifiers to segment V79 cells in PCM, a dataset was split into two subsets: a training dataset (25%) and a validation dataset (75%)(Skoczylas, Rakowski et al. 2011; Skoczylas, Rakowski et al. 2011). Cross-validation was also used to determine segmentation or feature extraction parameters (Seungil, Ker et al. 2011) in PCM. Tenfold cross-validation was used to evaluate the Adaboost classifier performance to classify cells as polarized/non-polarized, spread/non-spread and oriented/non-oriented in PCM microscopy, computing ROC curves (TPR versus FPR) (Theriault, Walker et al. 2012).

Other methods relying on a set of synthetic ground truth images have been proposed. For example, when segmenting fuzzy objects, a design of experiment (DOE) methodology was used to test two algorithms on several synthetic ground truth images having different signal-to-noise ratios and target to background features (image characteristics). The goal was to predict by multiple linear regression (MLR) the accuracy of segmentation relative to image characteristics for each algorithm (Zhou, Zhang et al. 1995).

2.4.9 Commercial products/softwares for image processing

2.4.9.1 Introduction

Advanced platforms were developed through the years to address specific issues and screening problems in biotechnology. A brief review of the different high throughput screening (HTS) platforms using microscopy is given by Rimon and Schuldiner (2011). Automated imaging equipment is available today, but software support is limited as most applications in the literature are implemented in manufacturers softwares or programmed in environments such as Matlab, LabView, IDL, Visual Basic and others (Berlage 2005). Commercial softwares were developed mainly for the pharmaceutical screening market by companies such as Cellomics, TTP LabTech, Evotec, Molecular Devices and GE Healthcare (Carpenter, Jones et al. 2006). The development of these products was aimed at studying mammalian cells and cellular features such as protein translocation, micronucleus formation, neurite outgrowth and cell counts. The bundling of commercial software with imaging hardware makes it difficult to test several algorithms for a new project, and the proprietary nature of the codes makes it difficult for researchers to modify them for a new application (Carpenter, Jones et al. 2006).

A critical step in each HTS project is image processing and data analysis. A brief review of the different software packages for image acquisition, image processing, image database management and machine learning was provided by Eliceiri et al. (2012). For a long time, the only flexible open-source biological image analysis software package was ImageJ/NIH (Carpenter, Jones et al. 2006). However, although ImageJ was very successful in several laboratories, it is directed towards the analysis of individual images instead of performing HTS work (Carpenter, Jones et al. 2006). As a consequence, many groups developed their own scripts, macros or plugins in order to perform automated cell image analysis in commercial softwares or freewares (MetaMorph, ImageProPlus, Matlab, Java, etc.) to identify, measure and track cells in time-lapse image sequences (Carpenter, Jones et al.

2006). The downside is that these custom programs are not modular, and combining several steps and changing settings require to interact directly with the source code (Carpenter, Jones et al. 2006), which is difficult for researchers not knowledgeable in those specific programming languages. Moreover, efforts spent by the laboratories to create these custom scripts are often lost after the initial experiments are completed, since each script is customized for a particular cell type, assay or image set (Carpenter, Jones et al. 2006). Also, testing multiple published algorithms for a new cell type or assay can be difficult, since programming each method can be impractical. Another main challenge is handling all the data from image processing following HTS experiments, and commercial image database management softwares such as SIMS (by BioImagene) or IQbase (by Media Cybernetics) are available (Berlage 2007).

Most image processing softwares or products try to achieve single-cell segmentation, but just a few can track cells from frame to frame. Automated cell tracking methods focus on associating cells across frames and struggle to provide proper information on the actual shape of the cells as they deform during their locomotion (Sacan, Ferhatosmanoglu et al. 2008). Most of the methods are developed by each research group in isolation, addressing a specific problem, but these scripts are not part of an integrated software environment or are closed source, thus limiting their applicability to new problems (Sacan, Ferhatosmanoglu et al. 2008). Few articles compare different cell tracking algorithms or softwares. One interesting paper was from Hand, Sun et al. (2009), in which five (5) cell tracking softwares were tested: Retrac (manual tracking), CellTrack (freeware), MTrack2 (ImageJ, freeware), Imaris and Volocity. Their paper showed that fluorescence microscopy instead of phase contrast microscopy improved the tracking efficiency of each software tested (Hand, Sun et al. 2009). A comparison between open source and commercial cell tracking algorithms is provided: commercial softwares tend to include more features (advanced image-processing techniques and 3D visualization) and technical support than their open-source counterparts (Hand, Sun et al. 2009). A wide-range of open-source softwares is available, ranging from basic numerical analysis of data about cells already segmented (e.g. Cell Motility) to more complex softwares (e.g. CellTracker) performing cell segmentation and tracking (Hand, Sun et al. 2009).

Because of the huge variety of products, scripts and commercial software packages described in the literature (open-source or commercial, hardware and/or software

platforms, etc.), a brief list of the products/softwares available for performing HTS via microscopy is provided in the next section.

2.4.9.2 Products/softwares available

A lot of different products/softwares are available commercially or are offered freely by several authors. These products/softwares can integrate several steps of cellular live-cell imaging and high-throughput screening, such as microscopy optics/hardware, image acquisition and image processing (steps shown in Figure 2-48). The different products or softwares found in the literature during this project (not an exhaustive list) are given in Appendix B. The extent of this list shows how much variety can be found in the literature.

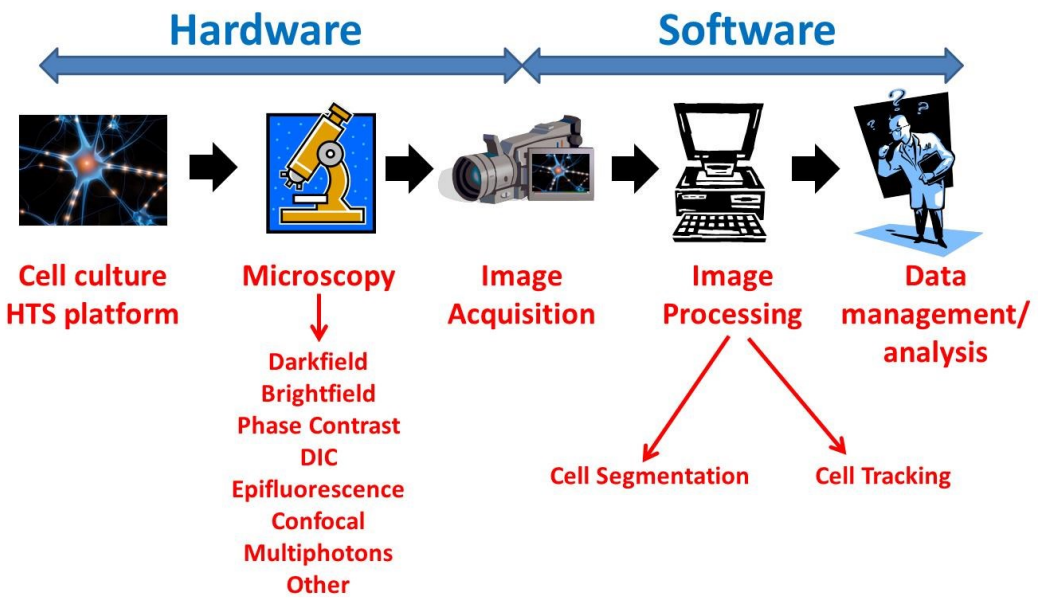


Figure 2-48 : Live-cell imaging/HTS steps for which several products/softwares offer a solution

Some products/softwares listed are described in more details, depending on the information found in the literature:

- Time Lapse Analyzer (TLA) is a customable Matlab-based software which can perform cell segmentation and tracking on fluorescence, brighfield, PCM and DIC microscopy images (Huth, Buchholz et al. 2011). In TLA, cell tracking is performed by cell centroid detection, data association and state estimation via a Kalman Filter. In DIC images, cell centroids are detected by deconvolution and local thresholding. In PCM images, edge-based cell detection and conditional erosion can be used

since there is a high contrast between the cell borders and the background. In brightfield images, denoising and image enhancement techniques (such as the Weiner filter) are used since there is a very low contrast between the cells and the background;

- CellProfiler® is a free open-source image analysis software that can address a variety of biological questions from a quantitative perspective, such as cell counts, size distributions, per-cell protein levels, cell/organelle shape, subcellular patterns of DNA and protein staining (Carpenter, Jones et al. 2006). It uses a script editor which combines functions in pipelines, and function parameters can be set in dedicated dialogue boxes (Huth, Buchholz et al. 2011). CellProfiler® provides an excellent support for cell classification and batch processing of large image files, but has limited support for time-lapse and three dimensional image stack analysis (Carpenter, Jones et al. 2006; Huth, Buchholz et al. 2011);

- CellTrack® is a cross-platform software package that incorporates a novel edge-based method for sensitive segmentation of cell boundaries which can achieve refined tracking results even under large cellular displacements and deformations (Sacan, Ferhatosmanoglu et al. 2008);

- Metamorph® is an acquisition and image processing software that incorporates classical techniques such as correlation matching, thresholding and morphological operations to perform segmentation (Cheng and Rajapakse 2009). The methods programmed have a limited use for images gathered in cell biology research due to overgeneralization (Cheng and Rajapakse 2009);

- New platforms using unconventional microscopy are more and more used. For example, the Holomonitor® uses holography to record 3D information about an object via interfering wave fronts from a laser (Mõlder, Sebesta et al. 2008). Using a conventional microscope, it is also possible to generate a phase map by taking a set of 3 brightfield images at adjacent focuses (around the optimal focal plane) and to solve the transport of intensity equation using published algorithms (Curl, Harris et al. 2004). To identify and track the position of embryo nuclei, it is possible to use

4D Normarski DIC microscopy via the SIMI BioCell® or the 3D-DIAS® platforms (Hamahashi, Onami et al. 2005).

Many software packages (such as ImageJ) provide basic functions (thresholding, morphological operations, etc.) and slightly more advanced programs (for image registration, background correction, etc.) that are generic and easily applicable to fluorescence image processing. However these packages generally do not provide the advanced programming capability required to develop from scratch a complete image processing algorithm tailored for a specific HTS application (generating a significant number of images and data) and for more complex cases involving for instance processing phase constrast microscopy images acquired under different conditions (high/low magnification, high/low confluence, uneven background, etc.). This explains why Matlab® (via the Image Processing Toolbox®) was preferred for this PhD project, in order to provide a maximum of programming flexibility to develop robust image processing solutions that will be able to analyze PCM images acquired under different conditions.

2.5 Multivariate data analysis

When analyzing multiple HTS experiments, multivariate analysis techniques are required because of the large datasets generated. Modern multivariate statistical methods (e.g. Principal Component Analysis, Partial Least Squares regression) provide powerful tools for analyzing the large sets of features and identifying combinations of features that allow samples to be discriminated and classified. However statistical results can be strongly influenced by the number of features used, since it can be easy to come up with overoptimistic classification results if hundreds of features are used for only a few dozen biomedical measurements (Rodenacker and Bengtsson 2003). This is the main reason why classification results and discriminant features need to be verified on an independent dataset, preferably by dividing the whole dataset into a calibration set and a validation set. If there are not enough biological measurements available to create two independent sets, the leave one out classification method (cross-validation) or the jackknifing approach can be used (Rodenacker and Bengtsson 2003).

2.5.1 Matrix of features

Image processing and analysis techniques typically yield a vector of features for each image. An overview of the different features available from image processing was presented in section 2.4.5. Additional features that are more relevant to HTS and cell

culture are the degree of confluence or the cell-covered surface (used in Chapter 4), cell counts and distributions of cellular morphological features (used in Chapter 6). When analyzing sets of images, it is possible to collect these features vectors into a **X** matrix. In the **X** matrix, each of the *n* rows correspond to an image and each of the *m* columns correspond to a single variable (feature value).The data matrix **X** can be seen as an ensemble of *n* points in a *m*-dimensional space. This *m*-space is difficult to visualize when there is more than 3 variables; however mathematically this space is similar to a two or three dimensional space (illustrated in Figure 2-45), and the same geometrical concepts for points, lines, planes, distances and angles can be applied (Wold, Esbensen et al. 1987). Since many features are correlated together, multivariate techniques are necessary.

2.5.2 Multivariate techniques

2.5.2.1 Data pre-treatment

The most common preprocessing step for multivariate analysis techniques is *auto-scaling*, in which each measured value is mean-centered by subtracting the mean value from each variable value, and is scaled to unit variance by dividing each mean-centered variable by its standard deviation. This method ensures that each variable has the same weight and the same prior importance in the analysis (in the absence of *a priori* knowledge about the relative importance of the variables).

When dealing with spectra and hyperspectral imaging, further processing can be performed to get rid of the noise and smooth the measured spectra (depending of the signal-to-noise ratio). Savitzky-Golay filters (Golay and Savitzky 1964) are commonly used to increase the signal-to-noise ratio of spectra while minimizing signal distortion. Several authors use spectra filtering as a pre-processing step, such as the second derivative (Gosselin 2010) or the fourth derivative (Polerecky, Bissett et al. 2009) to identify spectral peaks of interest.

2.5.2.2 Principal Component Analysis (PCA)

When the number of variables in **X** (i.e. columns) is large and when these are strongly correlated, Principal Component Analysis (PCA) can used to project the data points onto a lower dimensional space (i.e. latent variable) capturing most of the variance in the dataset **X**. PCA is also known under other names: singular value decomposition (SVD) in numerical analysis, Karhunen-Loève expansion in electrical engineering, Hotelling transformation in image analysis, correspondence analysis and factor analysis (FA) in some scientific fields (Wold, Esbensen et al. 1987).

PCA provides an approximation of a data matrix **X** in terms of the outer product of two matrices (**T** and $\mathbf{P}^T$), capturing essential data patterns in **X**. The matrix **T** provides an image of the dominant objects patterns and $\mathbf{P}^T$ shows the variables patterns (Wold, Esbensen et al. 1987). The projection of **X** down to an *A*-dimensional subspace is performed by projecting the data points onto the projection matrix ($\mathbf{P}^T$), giving the objects new coordinates in this space (**T**) . The columns in **T** are called the score vectors and the rows in $\mathbf{P}^T$ are called the loading vectors. The loadings vectors provide the direction coefficients for the principal components (PC) hyperplane. Both the loadings and the scores vectors are orthogonal. The deviations between the projections and the original coordinates are called *residuals* (Wold, Esbensen et al. 1987). In the *m* space, PCA can be seen as a projection of *n* data points onto a lower *A* dimensional space. Figure 2-49 shows a projection of data point x_i onto $\mathbf{p}_1$ ($t_{i,1}$).

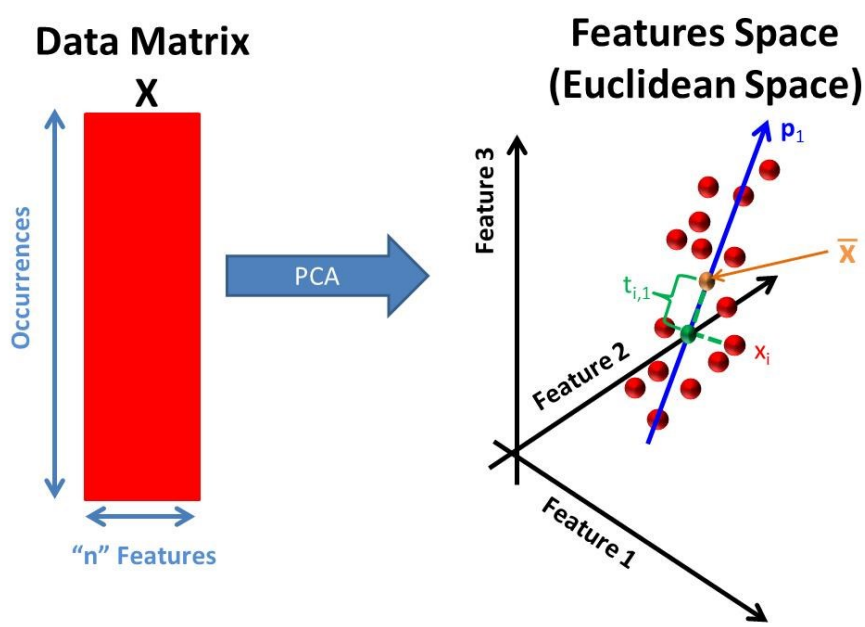


Figure 2-49 : One component PCA model capturing the variation in a 3 dimensions features space in which a data point x_i is projected onto p_1 ($t_{i,1}$) (adapted from Wold, Esbensen et al. (1987))

In matrix notation, the **X** matrix is decomposed into principal components according to the following equation (Geladi and Grahn 1996):

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{2-76}$$

In which **E** is a residuals matrix. The relationship between these matrices is illustrated in Figure 2-50.

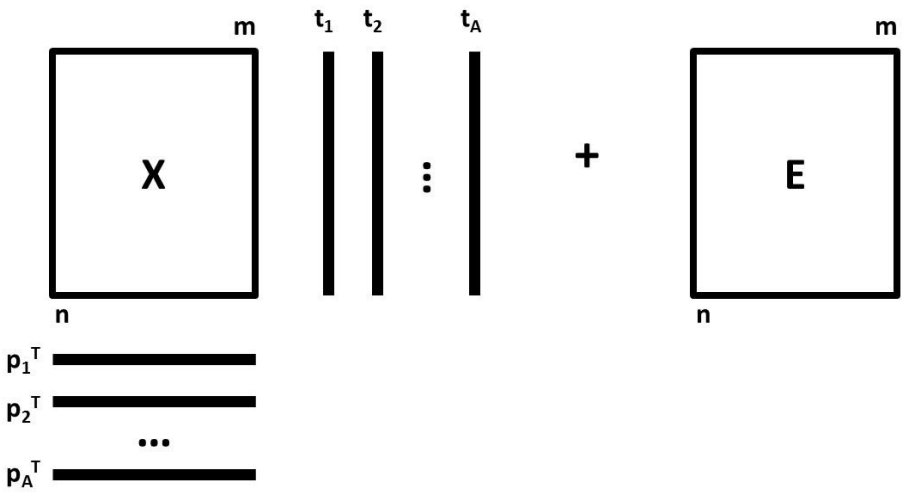


Figure 2-50 : Size of the different matrices involved in PCA (adapted from Wold, Esbensen et al. (1987))

From the first component, each component added is assigned loadings to capture most of the remaining variance in **X** while being orthogonal to the previous one. In PCA, the score and loading vectors corresponding to the largest eigenvalues explain the most variance in the **X** matrix and are written in order of descending eigenvalues (Wold, Esbensen et al. 1987). Plotting the columns of **T** against each other shows the configuration of the objects (patterns) in **X**. The last principal components often capture noise and are neglected (Geladi and Grahn 1996).

The PCA technique can be used for data simplification, data reduction, data modeling, outliers detection, variable selection, classification, prediction and unmixing (Wold, Esbensen et al. 1987). PCA decomposition is particularly useful for spectral analysis (e.g. hyperspectral image analysis), since there is a strong correlation between intensities measured at adjacent wavelengths (Geladi and Grahn 1996). Using PCA, an hyperspectral image can be decomposed into a summation of score images that can be analyzed to detect textural patterns.

One of the main algorithms to compute PCA is the Nonlinear Iterative Partial Least Squares (NIPALS) algorithm which computes the principal components sequentially (Geladi and Kowalski 1986). The NIPALS algorithm is well described in the literature

(Wold, Esbensen et al. 1987). This algorithm computes $\mathbf{t}_1$ and $\mathbf{p}_1$ from the **X** matrix and subtracts the outer product of these two matrices from **X** to compute a first residual matrix $\mathbf{E}_1$. The first residual matrix $\mathbf{E}_1$ is used to compute $\mathbf{t}_2$ and $\mathbf{p}_2$, and these steps are repeated until the last principal component is computed (**X** is deflated after the computation of each principal component). The conventional way to determine the number of components is by cross-validation (Wold 1978), in which latent components are added as long as they improve model predictions (see section 2.5.2.5). The model predictive ability can be evaluated using the Q^2 coefficient (defined in equation (2-88)), and components can be added as long as Q^2 increases. Several other criteria are also available in the literature. Another common approach is the use of a scree plot, in which one plots the value of each successive eigenvalue in decreasing order (Jackson 1993). The larger the eigenvalue, the more variation is captured in the **X** matrix. The smaller eigenvalues tend to align themselves along a straight line, and the point where the first few eigenvalues depart from the line highlights the difference between important latent components and trivial components (Jackson 1993).

2.5.2.3 Partial Least Squares/Projection to Latent Structures (PLS)

Partial Least Squares projection to latent structures (PLS) is a regression method that relates two data matrices, **X** and **Y**, by a linear multivariate model, and goes beyond traditional regression since it can also model the latent structures of **X** and **Y** (Wold, Sjostrom et al. 2001). PLS tackles collinearity problems as PCA does. However PLS derives its usefulness from its ability to analyze data with many noisy, collinear and even incomplete **X** and **Y** variables, something that can hardly be done by traditional Multiple Linear Regression (MLR). PLS provides a model of **X** in terms of a bilinear projection, assuming that there may be parts of **X** that are not related to **Y** (usually these parts include noise and/or other unrelated patterns) (Wold, Sjostrom et al. 2001). This makes PLS models more tolerant to noise than MLR models, which is a huge advantage when correlating for example measured spectra (e.g. near-infrared spectra) to specific samples characteristics (e.g. chemical composition).

While PCA maximizes the variance captured in **X**, PLS maximizes the covariance between **X** and **Y**. A PLS model consists of outer relations (latent components for **X** and **Y** blocks individually) with an inner relation linking both blocks (Geladi and Kowalski 1986). The outer relation for **X**, the outer relation for **Y** and the relation for computing **T** scores

from **X** are provided in equations (2-77) to (2-79) (Wold, Geladi et al. 1987), assuming that **X** scores are good predictors of **Y** :

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{2-77}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} \tag{2-78}$$

$$\mathbf{T} = \mathbf{X}\mathbf{W}^{*} = \mathbf{X}\mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1} \tag{2-79}$$

In which **T** are the **X** scores, **P** are the **X** loadings, **U** are the **Y** scores, **Q** are the **Y** loadings and **E** and **F** are residuals matrices. **W** is a matrix of weights that provides the combination of **X** that are the most predictive of **Y**, and are used to calculate the scores **T** for each component while ensuring an orthogonal latent space for both **X** and **Y**. To compute the **T** scores, a transformed weight matrix (**W***) is used (modified **X** loadings) (Wold, Sjostrom et al. 2001). The **W** matrix provides weights that combine the **X** variables (first component) or the residuals of the **X** variables (for subsequent components) to compute the **T** scores (Umetrics 2005). The **W*** matrix provides the weights to compute the **T** scores directly from the original **X** matrix for all components.

The inner relation can be seen in a graph of the **Y** block scores, **U**, against the **X** block scores, **T**, for each component (Geladi and Kowalski 1986). The simplest inner relation model is a linear one (Geladi and Kowalski 1986). The inner relation linking both the **X** and **Y** blocks is expressed by the following equation (Wold, Geladi et al. 1987):

$$\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{H} \tag{2-80}$$

In which **B** is a diagonal matrix relating **X** and **Y** scores, and **H** is a residuals matrix. The **B** matrix incorporates the slopes between the **U** and **T** scores. Assuming that the **X** scores are good predictors of **Y**, it is possible to consider that **U** = **T** and that the **B** matrix is the identity matrix (**I**). As a consequence, both **X** and **Y** spaces are linked through a unique score space **T**. The matrices involved in a PLS model with *A* principal components is illustrated in Figure 2-51.

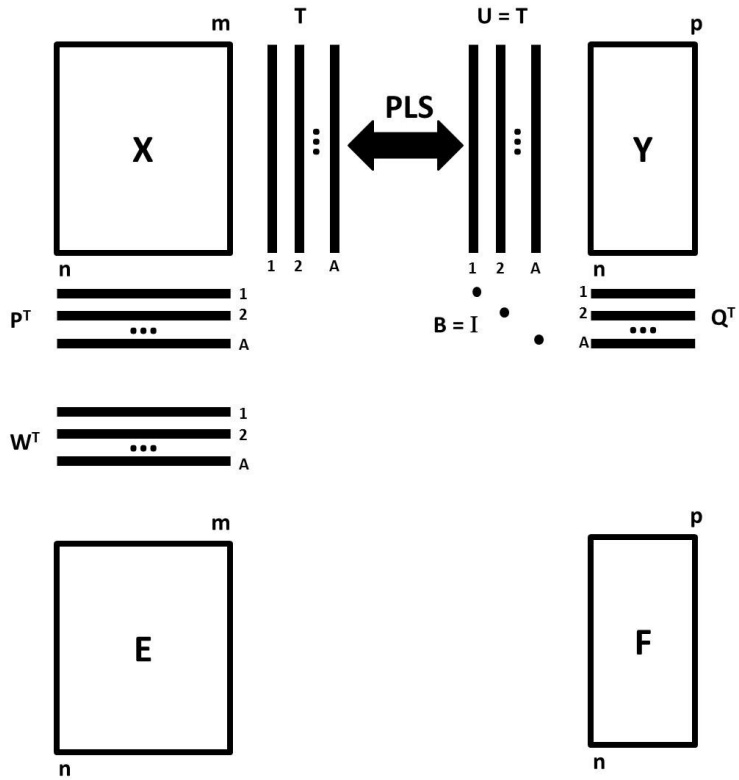


Figure 2-51 : Size of the different matrices involved in PLS (adapted from Wold, Geladi et al. (1987) and from Wold, Sjostrom et al. (2001))

To export the PLS model (as done in Chapter 6 for computing morphological features without the PLS toolbox) and to predict the **Y** matrix variables from features in the **X** matrix, the following equations are used:

$$\mathbf{Y} = \mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{F} \tag{2-81}$$

$$\widehat{\boldsymbol{\beta}} = \mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{Q}^{\mathrm{T}} = \mathbf{W}^{*}\mathbf{Q}^{\mathrm{T}} \tag{2-82}$$

In which $\widehat{\boldsymbol{\beta}}$ is a matrix providing linear coefficients of regression.

Standard algorithms to compute PLS are the NIPALS iterative algorithm (Wold, Sjostrom et al. 2001) and the SIMPLS algorithm (Dejong 1993). The number of components to be used (*A*) is a very important parameter, as it is possible to calculate as many PLS components as the rank of the **X** block or the **Y** block (A_{max} = min(rank(**X**,**Y**)). However not all of them are normally used, as some of the components associated with smaller eigenvalues will only capture noise (Geladi and Kowalski 1986). It is thus common to leave out a significant number of components because they can carry collinearity problems.

However nonlinear models require extra components to describe nonlinearities (Geladi and Kowalski 1986).

To interpret PLS models, plotting w's or w*'s against q's for one principal component against another can help to understand the latent space model structure (Wold, Sjostrom et al. 2001). This plot is termed the loadings bi-plot. It shows how the **X** variables combine to form the scores $\mathbf{t}_i$, with the most important for the ith component falling far from the origin on the ith axis (Wold, Sjostrom et al. 2001). In a similar manner, **Y** variables well modeled by the ith component have large $\mathbf{q}_i$ loadings and fall far from the origin on the ith axis. **X** and **Y** variables correlated together will fall near each other in the same region of the loadings bi-plot.

A summary of the importance of an **X** variable to model both **Y** and **X** is provided by its VIP (variable importance for the projection) (Wold, Sjostrom et al. 2001). The equation to compute the VIP score associated to the kth variable in the **X** matrix for *A* components is provided in the following equation (Eriksson, Kettaneh-Wold et al. 2001; Gosselin, Rodrigue et al. 2010):

$$\mathrm{VIP_k} = \sqrt{\mathrm{K}\sum_{a=1}^{A}\left[\left(\mathbf{q}_a{}^2\mathbf{t}_a{}^T\mathbf{t}_a\right)\left(w_{ak}{}^2/\|\mathbf{w}_a\|^2\right)\right]/\sum_{a=1}^{A}\left(\mathbf{q}_a{}^2\mathbf{t}_a{}^T\mathbf{t}_a\right)} \qquad (2\text{-}83)$$

In which w_{ak} is the weight for the kth variable for the ath component and $\mathbf{t}_a$, $\mathbf{w}_a$ and $\mathbf{q}_a$ are the ath column vectors of the **T**, **W** and **Q** matrices. Since the average of VIP scores is equal to 1, the X-variables with a VIP larger than 1 are the most relevant for explaining **Y** variables (Umetrics 2005).

A common problem in multivariate analysis is dealing with missing values, and most techniques relying on the construction of a latent space (namely PCA and PLS) can provide an estimate of the missing values without influencing too much the end result. For example, the NIPALS algorithm can automatically account for missing values by iteratively substituting the missing values with predictions made by the PLS model, thus replacing missing values with estimates that have zero residuals and no influence on the scores and the loadings determination process (Wold, Sjostrom et al. 2001). Holes in a **X** data matrix can also be filled using PCA (Wold, Esbensen et al. 1987).

To detect objects or occurrences that are outliers, X-residuals (in the **E** matrix) can be used. The standard deviation of the X-residuals for each row in the residuals matrix **E** can

be used, since it is proportional to the distance between the data point (occurrence) and the model plane in the X-space (Wold, Sjostrom et al. 2001), and is often termed DModX (distance to the model in the X-space). A DModX larger than around 2.5 times the overall standard deviation of the X-residuals (corresponding to a F-value of 6.25) often indicates that the object or occurrence is an outlier (Wold, Sjostrom et al. 2001).

The goodness of fit of the PLS model is given by R^2 and Q^2 (cross-validated R^2) for each **Y** variable (Wold, Sjostrom et al. 2001). The R^2 provides an idea of how well the model explains the data in the training set as Q^2 provides an idea of how well the model predicts new observations in the validation set. If R^2 and Q^2 are acceptable, the model is interpretable, and it is possible to fine tune it by deleting unimportant **X** variables associated to small regression coefficients and low VIP values (Wold, Sjostrom et al. 2001).

2.5.2.4 Multivariate Image Analysis (MIA)

Multivariate image analysis (MIA) essentially consists of applying PCA to a multivariate image, and using the various PCA plots to explore the information contained in the image. Prior to applying PCA, the multivariate image (i.e. a three-way array of data or cube) need to be reorganised into a matrix where the light intensities at different wavelengths corresponding to each pixels are stored row-wise in the **X** matrix. This so-called image unfolding operation is illustrated in Figure 2-52 using a RGB color image (Houses RGB image, Kingdom and Olmos). The data cube associated with this RGB image can be splitted along one of the spatial axis into k=512 slices ($\mathbf{X}_i$). In order to perform this computation efficiently on several images, a kernel algorithm is used (Geladi and Grahn 1996).

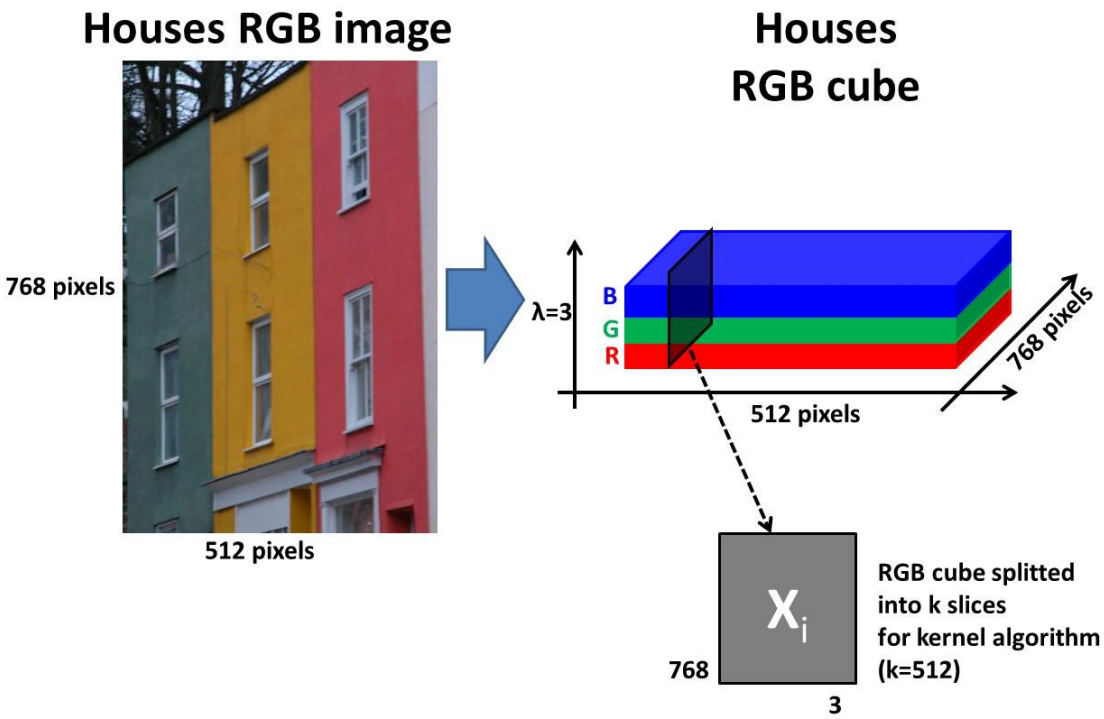


Figure 2-52 : Houses RGB image (Kingdom and Olmos) and corresponding RGB cube. It is possible to analyze the RGB cube slice by slice (X_i).

For a cube made of k slices $\mathbf{X}_i$, the kernel matrix **Z** (variance-covariance matrix) is computed as follows:

$$\mathbf{Z} = \sum_{i=1}^{k} \mathbf{X}_i{}^T \mathbf{X}_i \qquad (2\text{-}84)$$

To compute the loadings, SVD is applied on the **Z** matrix (which has dimensions λ by λ). The loadings will indicate the relative contributions of each wavelength to each component. The loadings of each principal component of the Houses RGB image are shown in Figure 2-53. The first component ($\mathbf{p}_1$) captures most of the variance (92.4%) as the components $\mathbf{p}_2$ and $\mathbf{p}_3$ capture respectively 6.1% and 1.5% of the variance in the data cube.

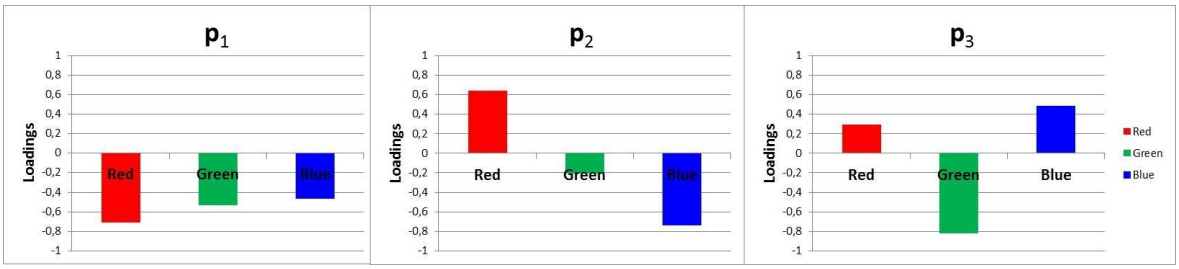


Figure 2-53 : p_1, p_2 and p_3 loadings from MIA performed on the Houses RGB image (Kingdom and Olmos).

From the loadings, score images can be computed (**T** = **XP**) for each image in the dataset and each component to show how the information captured by each component is located spatially. $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ score images for the Houses RGB image are provided in Figure 2-54. Note that the score images were rescaled between 0 and 255 to enhance the contrast.

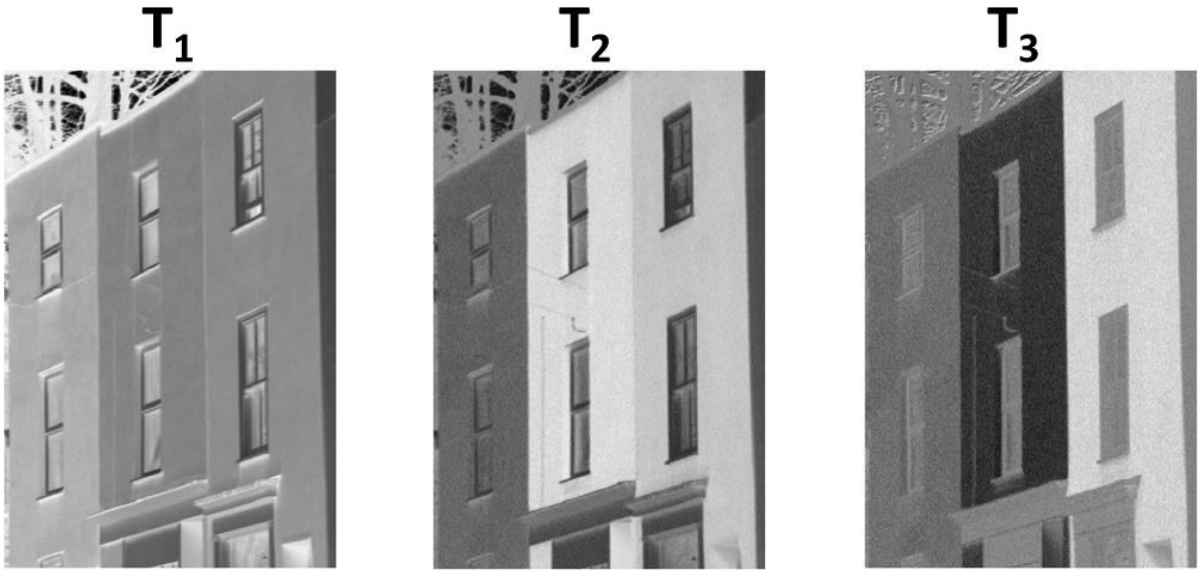


Figure 2-54 : T_1, T_2 and T_3 score images from MIA performed on the Houses RGB image (Kingdom and Olmos).

The first component ($\mathbf{T}_1$ computed from $\mathbf{p}_1$) captures variations in light intensities across the image (all 3 loadings in $\mathbf{p}_1$ have the same sign) and represent a grey level version of the image. The second component ($\mathbf{T}_2$ computed from $\mathbf{p}_2$) shows the contrast between the

red and the blue channels associated with the houses in the image. The third component ($\mathbf{T}_3$ computed from $\mathbf{p}_3$) shows the contrast between the green channel and the other channels. Usually the last component captures noise (as in this case it explains only 1.5% of the variance in the RGB image).

Score density histograms are used to see how pixels cluster together in the scores space as they are associated to similar spectral/color patterns. Usually the scores for the ath component are rescaled using the minimum and maximum score values ($t_{a,min}$ and $t_{a,max}$), and the range of rescaled scores is divided into bins using a rectangular grid. For the ath latent component, the scores ($\mathbf{T}_a$) are rescaled between 0-255 (min-max scaling), rounded and binned to create a new score matrix ($\mathbf{T}_{a,new}$) which assigns a bin # (1 to 256) to each score, as shown in the following equation:

$$\mathbf{T}_{a,new}(\text{bin \#}) = 1 + \text{round}\left[\left(\left(\mathbf{T}_a - t_{a,min}\right)/\left(t_{a,max} - t_{a,min}\right)\right) * 255\right] \quad (2\text{-}85)$$

Once bins are assigned to each score for each latent component, it is then possible to create a 2D density histogram (256 bins by 256 bins) by counting the number of scores that fall into each 2D bin (for two specific components), using bin numbers as coordinates. A colormap is then applied to highlight areas which have a higher scores density (usually a brighter color). The $\mathbf{T}_1/\mathbf{T}_2$, $\mathbf{T}_1/\mathbf{T}_3$ and $\mathbf{T}_2/\mathbf{T}_3$ score density histograms for the Houses RGB image are provided in Figure 2-55. Note that the origin of the score density histograms (bin 1,1) is located in the top-left corner.

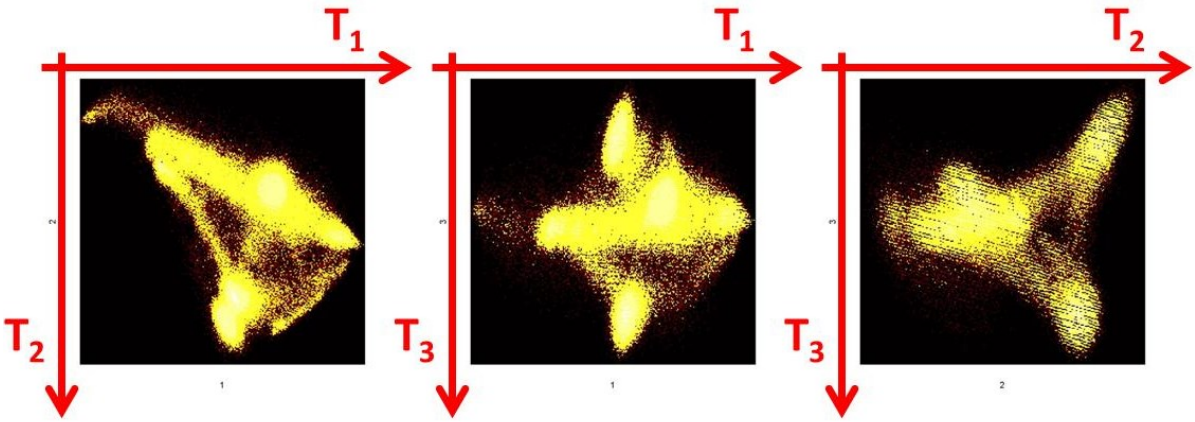


Figure 2-55 : T_1/T_2, T_1/T_3 and T_2/T_3 score density histograms from MIA performed on the Houses RGB image (Kingdom and Olmos).

To explore the image information, one can select groups of pixels (clusters) within the score density histogram and overlay the selected pixels on the original image. This operation is called the masking procedure. The masks segment pixels in the score space that have similar spectral characteristics (here colors). The traditional way to extract features from MIA is to use masks in the scores space and to count the number of pixels under each mask or to track the pixels back into the original image for segmentation (Duchesne, Liu et al. 2012). There are several masking methods available: trial-and-error, method using **Y** data to create parallel or curved mask bins (Yu and MacGregor 2003) or methods relying on automatic supervised classifiers such as SVM (Liu, Bharati et al. 2005). The number of masks to use can be determined by trial-and-error or by an automated procedure combing unsupervised and supervised classifiers to extract the number of classes in an iterative manner (Noordam, van den Broek et al. 2005). To explore the $\mathbf{T}_1/\mathbf{T}_2$ score density histogram associated with the Houses RGB image, a trial-and-error method is used to manually draw several masks to explore the scores space. The results for several masks are shown in Figure 2-56.

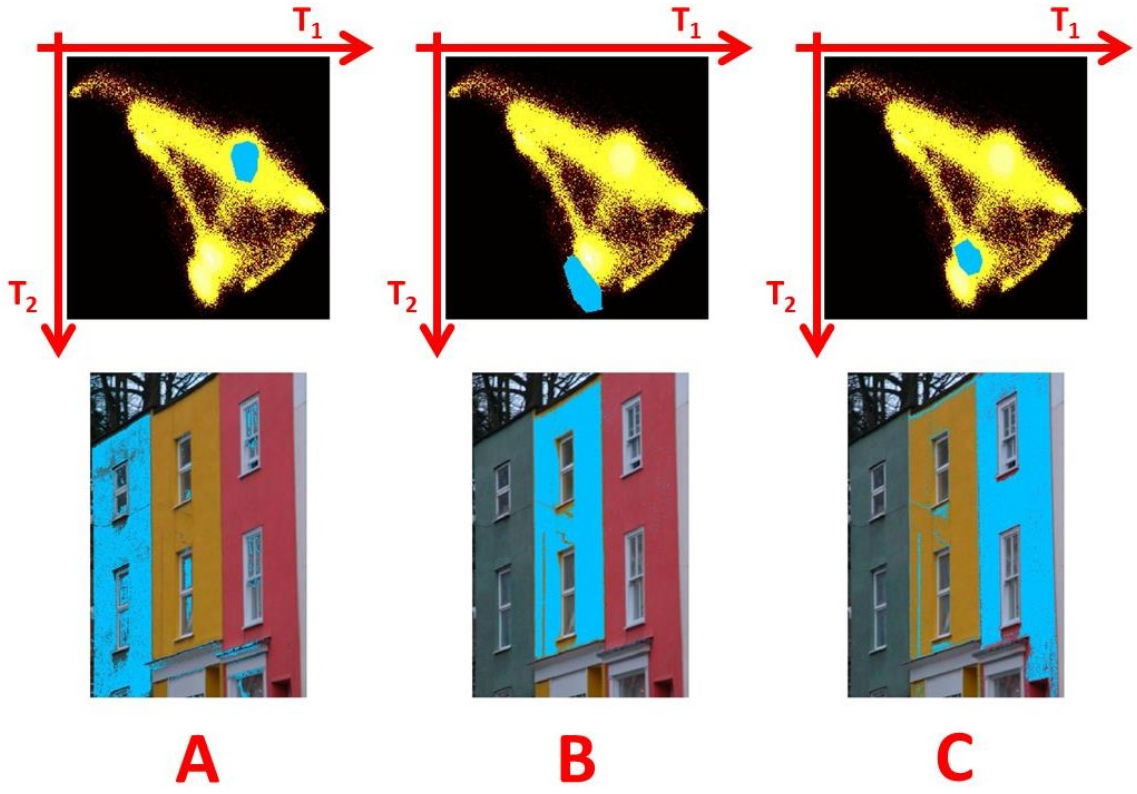


Figure 2-56 : T_1/T_2 masking results (selected regions in cyan) from MIA performed on the Houses RGB image (Kingdom and Olmos).

In Figure 2-56, since scores tend to distribute along $\mathbf{T}_1$ according to the light intensities across the image, it is very hard to select pixels clusters relative to their color information. However, along $\mathbf{T}_2$, it is possible to select three groups of pixels, associated with the green house (Figure 2-56A), the yellow house (Figure 2-56B) and the red house (Figure 2-56C). When looking at the $\mathbf{p}_2$ (see Figure 2-53), the red channel has a high positive loading, explaining why pixels associated to the red and yellow houses end-up in the lower portion of the score density histogram (high $\mathbf{T}_2$ scores for the red and yellow houses pixels). Since the yellow houses pixels have a higher light intensity than the the red houses pixels, they end-up slightly lower in the score density histogram (stronger $\mathbf{T}_2$ scores). Since the green house pixels have strong blue channel intensities and since the blue channel loading in $\mathbf{p}_2$ (see Figure 2-53) is strongly negative, these pixels will have low $\mathbf{T}_2$ scores and will end-up in the upper portion of the score density histogram.

The kernel algorithm used can also handle several images at the same time (see Chapter 5). Each data cube can be splitted into slices and the PCA can be computed by performing a SVD on the variance-covariance matrix **Z**. The **Z** matrix is thus computed as follows for *m* images, for $\mathbf{X}_i$ slices belonging to image *j*:

$$\mathbf{Z} = \sum_{j=1}^{m} \sum_{i=1}^{k} \mathbf{X}_{i,j}{}^{T} \mathbf{X}_{i,j} \qquad (2\text{-}86)$$

Several MIA custom Matlab® scripts were developed for this PhD project. A good GUI already available was developed in Matlab® by the McMaster Advanced Control Consortium (MACCMIA, Dunn (2010)). The Houses RGB image was submitted to MACCMIA to validate the Matlab® scripts developed for this project. Both approaches provided the same results for the Houses RGB image. However it is important to note that the MACCMIA GUI can only handle one image at a time, as our Matlab® MIA custom scripts can handle a whole dataset having multiple images.

2.5.2.5 Performance of multivariate models

When analyzing chemical data to build a model between independent variables (**X**) and dependent variables (**Y**), two steps are required: calibration and validation (Geladi and Kowalski 1986). Usually the data matrix is split into two subsets: a training set (calibration) and a test set (validation). Model parameters are adjusted using the training set in order to minimize for each **Y** variable the sum of the square differences between the predicted dependent variables ($\hat{\mathbf{y}}$) and the actual values of the dependent variables (**y**), termed SSE.

Once the model parameters are fitted, the test set is used to assess the model predictive performance on new data points that were not used for calibration. Calculating the degree of fit for both the calibration and validation sets (R^2_{cal} and R^2_{val}) is a good method to make sure that a model has a good performance in prediction. It also makes sure that overfitting (using too many parameters) is avoided, which would be detrimental to the performance of the multivariate models. Overfitting can usually be detected when a high R^2_{cal} is obtained for the calibration set while a low R^2_{val} is obtained for the validation set. In that sense a model can be considered as robust if the model parameters do not change substantially when new calibration samples are added from the total population (Geladi and Kowalski 1986).

When dealing with numerous and correlated X-variables and a limited number of data points, there is a substantial risk of overfitting (Wold, Sjostrom et al. 2001), i.e. building a well fitted model with little predictive power. A test needs to be performed to assess the predictive capacity of each PCA or PLS component so just the right number of latent components is used (Wold, Sjostrom et al. 2001). Cross-validation (CV) is a practical and reliable method to assess the predictive significance of each latent component. In CV parts of the data is kept out of the model calibration process: the kept out data is then predicted by the calibrated model, and the predicted values are compared to the actual values (Wold, Esbensen et al. 1987). The squared differences between the predicted ($\hat{\mathbf{y}}$) and the actual ($\mathbf{y}$) values are summed and used to compute the prediction sum of squares (PRESS) or the root mean square error of cross-validation (RMSECV). PRESS can be seen as a metric representing the lack of prediction accuracy (Geladi and Kowalski 1986).

The root mean square error of cross-validation can be defined as (Gosselin 2010):

$$\mathrm{RMSECV} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n}\left(f_{cv,i}\right)^2}{n}} = \sqrt{\frac{\mathrm{PRESS}}{n}} \qquad (2\text{-}87)$$

It is possible to define R^2 and Q^2 for each **Y** variable with the following equations (Wold, Sjostrom et al. 2001):

$$Q^2 = 1 - \mathrm{PRESS}/\mathrm{SST} \qquad (2\text{-}88)$$

$$R^2 = 1 - \mathrm{SSE}/\mathrm{SST} \qquad (2\text{-}89)$$

In which SST represents the total variance of each **Y** variable (having *n* points), which is computed for each *n* x 1 variable vector **y** as follows:

$$\mathrm{SST} = (\mathbf{y} - \bar{\mathbf{y}})^{\mathrm{T}}(\mathbf{y} - \bar{\mathbf{y}}) \tag{2-90}$$

In which $\bar{\mathbf{y}}$ is a *n* x 1 vector with all values equal to the mean of **y**:

$$\bar{\mathrm{y}} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{2-91}$$

PRESS or RMSECV is calculated for each number of components (for up to 10 or 15 components), and the PLS model having the lowest PRESS is used (Wold, Sjostrom et al. 2001), avoiding the use of too many components (avoiding overfitting). The evolution of PRESS against the number of components (*A*) can be represented graphically (see Figure 2-57).

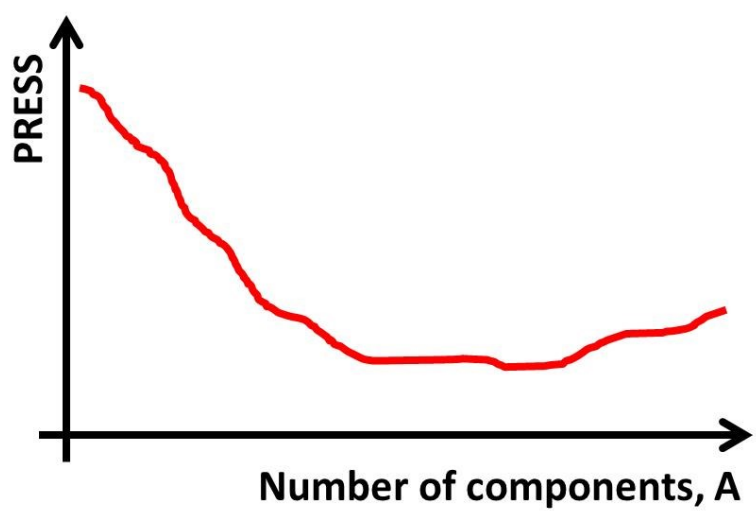


Figure 2-57 : PRESS as a function of the number of components *A* (adapted from Geladi and Kowalski (1986))

Other methods are exploited in the literature to look at the prediction capacity of multivariate models such as bootstraping and jackknifing (Duchesne and MacGregor 2001). Jackknifing can estimate standard errors and confidence intervals directly from the data (Wold, Sjostrom et al. 2001). Bootstrapping uses the residuals to simulate a large number of data sets similar to the original data to study the distribution of model parameters (Wold, Geladi et al. 1987).

Chapter 3. Materials and Methods

3.1 Cell culture

Myoblasts from a primary cell line were cultivated over several passages in a medium containing serum (MB-I + 15% FBS) and in a serum-free medium (LOBSFM, Parent (2009)). Cells were grown in 6 wells multiwell plates for imaging as well as in 25 cm^2 T-flasks to expand them through passages at 5% CO_2 and at 37.5 ^{0}C (in a controlled incubator). When performing live-cell imaging experiments, myoblasts were kept inside the microscope in an environment containing 5% CO_2 and at a temperature of 37.5 ^{0}C. The growth medium was changed every 2-3 days. The protocols used to cultivate myoblasts were inspired by Parent (2009) and are given in Appendix C. The culture media used were a serum-supplemented medium (SSM, described in Table C-1) and a serum-free medium called LOBSFM (SFM, composition provided in Table C-2).

3.2 Microscopy

The microscope enclosure shown in Figure 3-1 was used for this PhD project. Phase contrast and hyperspectral images were captured by an IX-81 Olympus microscope equipped with a moving stage. The IX-81 inverted microscope is automated with the Metamorph® software (Molecular Devices™) through the microscope's controller. A Qimaging® monochrome camera (model RETIGA 2000R) attached to the side port of the microscope (Figure 3-1) was used for brighfield, phase contrast and epifluorescence microscopy. The fitting of the microscope side port can also be used to attach one of the three hyperspectral cameras (UV, VIS and NIR) already available from a previous project (Gosselin 2010).

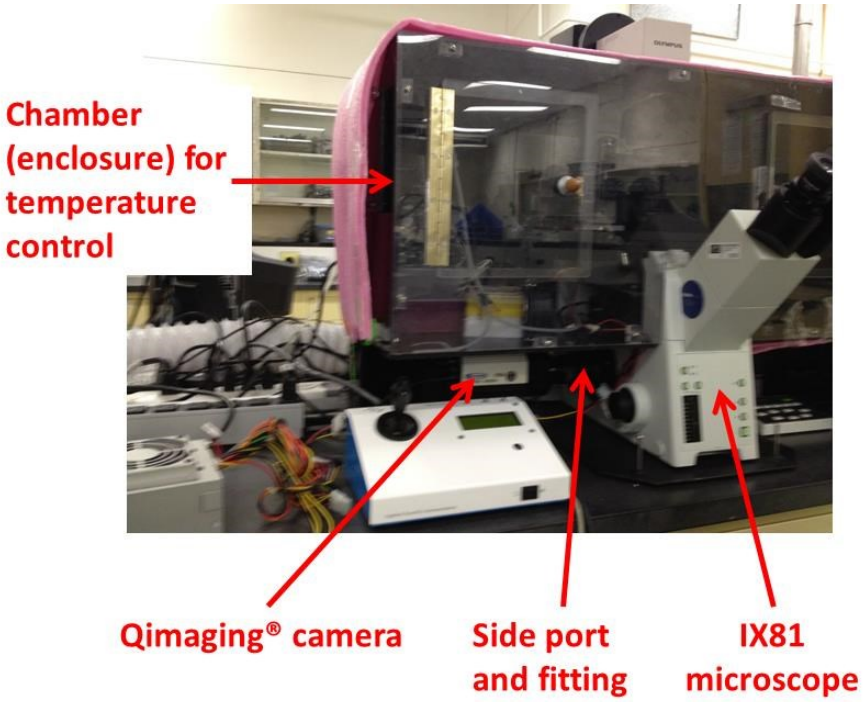


Figure 3-1 : Qimaging® monochrome camera attached to the side port of the IX81 Olympus™ microscope

The automated stage and the controller were purchased from ASI™ and can move in the x and y directions with great accuracy. The ASI™ stage is controlled with the MS-2000 controller, with a resolution for both axis of 22 nm (rotary encoder step for a standard lead screw pitch of 6.35 mm). It can move the stage by multiples of the mimimum speed (0.17 µm/s). The microscope is equipped with two lighting systems: a halogen light source located at the top, used for brightfield and phase contrast microscopy, and a UV light source located at the back (Ploem illumination) for epifluorescence microscopy. The condenser of the microscope is also automated, allowing to choose filters and rings for brightfield and phase contrast microscopy. For epifluorescence microscopy, a carrousel with different excitation and emission dichroic filters is incorporated into the body of the microscope and can be controlled as well. The objectives are mounted on an automated turret. The focal plane adjustment (by moving up and down the objectives) is performed via Metamorph® for focus adjustment and for imaging multiple focal planes (for suspended cells). The enclosure was custom made in the Chemical Engineering Department (Université Laval) in a previous project (Leysi-Derilou 2011). This enclosure is made of plastic with metal screws so it can be dismantled if needed.

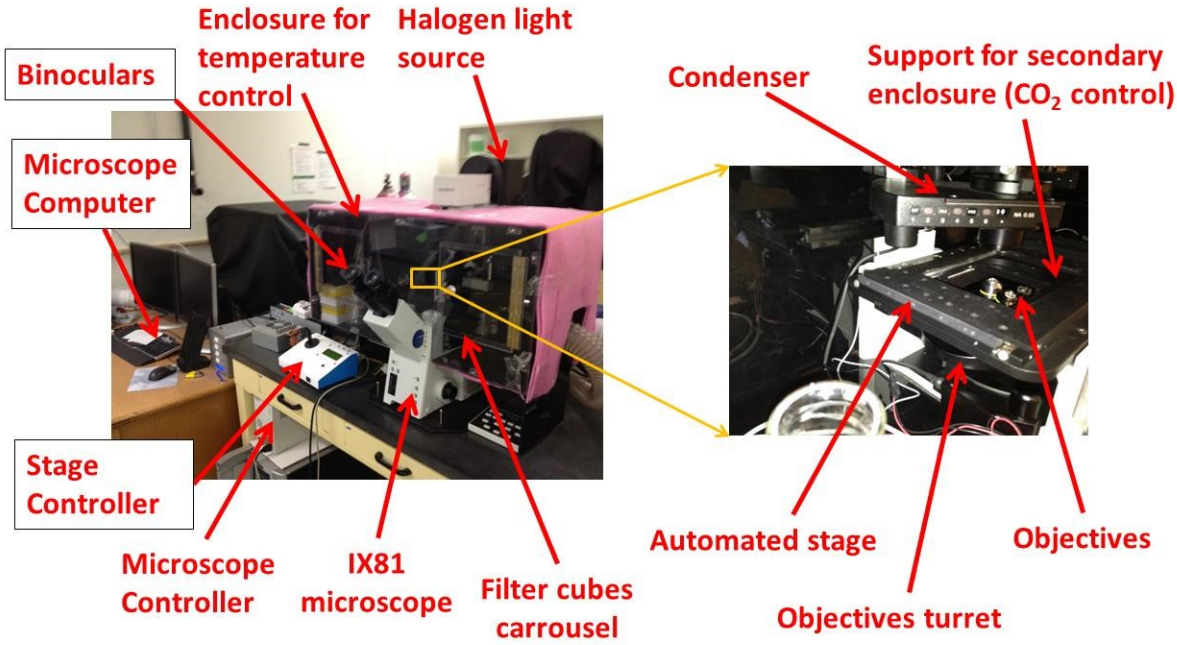


Figure 3-2 : IX81 Olympus™ microscope with enclosure

Several objectives can be mounted on the microscope, but the space available between the objectives turret and the automated stage frame is limited. Brightfield and phase contrast (with a ring incorporated) objectives providing a magnification of 10X, 20X and 40X are available. The resolution of the microscopic images that can be achieved with the objectives available was roughly assessed by taking images of a hemacytometer (in PCM). The hemacytometer images are shown in Figure 3-3. Note that the images are slightly slanted because the hemacytometer was fixed manually with tape on a multiwell plate holder placed on the microscope automated stage.

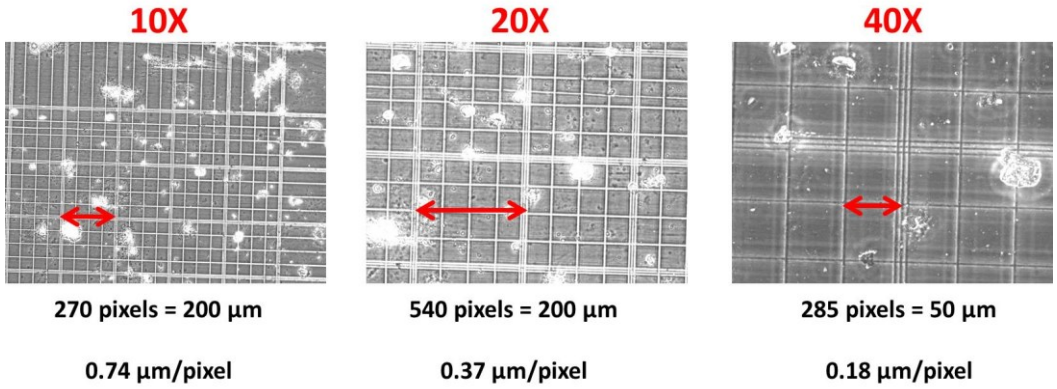


Figure 3-3 : IX81 Olympus™ microscope resolution in PCM from hemacytometer images.

Experimental resolutions (measured using an hemacytometer) are provided in Table 3-1.

Table 3-1 : Experimental resolutions for each microscope objective

Objectives	N.A.	Measured resolution with hemacytometer (µm/pixel)
10X	0.25	0.74
20X	0.45	0.37
40X	0.6	0.18

For temperature control (Figure 3-4), an air heater and a fan (Air Therm ATX®, World Precision Instruments™) are used to heat and flow the air through the enclosure and maintain a constant temperature via a PID controller (the temperature is measured via a temperature probe installed in the CO_2 secondary enclosure). A CO_2 controller (Figure 3-4) mixes air and pure CO_2 from bottles in specific proportions to obtain the proper concentration. The air/CO_2 mix is sparged in water in an Erlenmeyer (to saturate the gas with water) before being sent to the CO_2 enclosure (Figure 3-5). This was done to make sure that the cells will grow in a proper atmosphere (with the appropriate CO_2 concentration) while minimizing culture medium evaporation (since the mix is saturated with water).

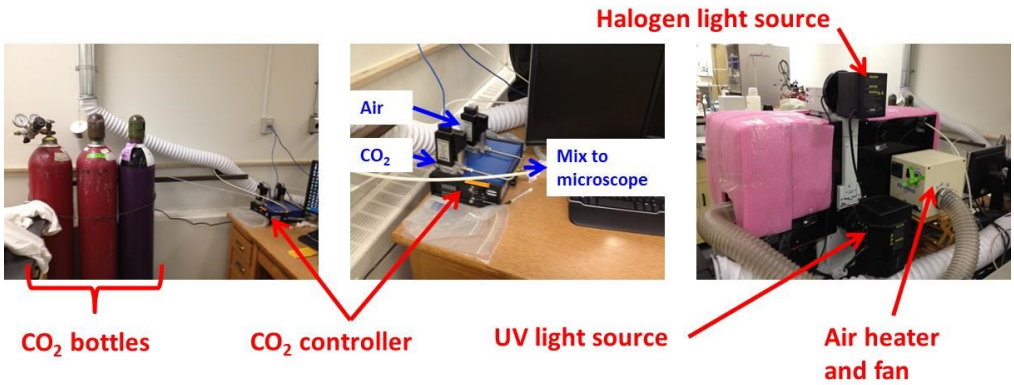


Figure 3-4 : CO_2 and temperature control of the experimental set-up

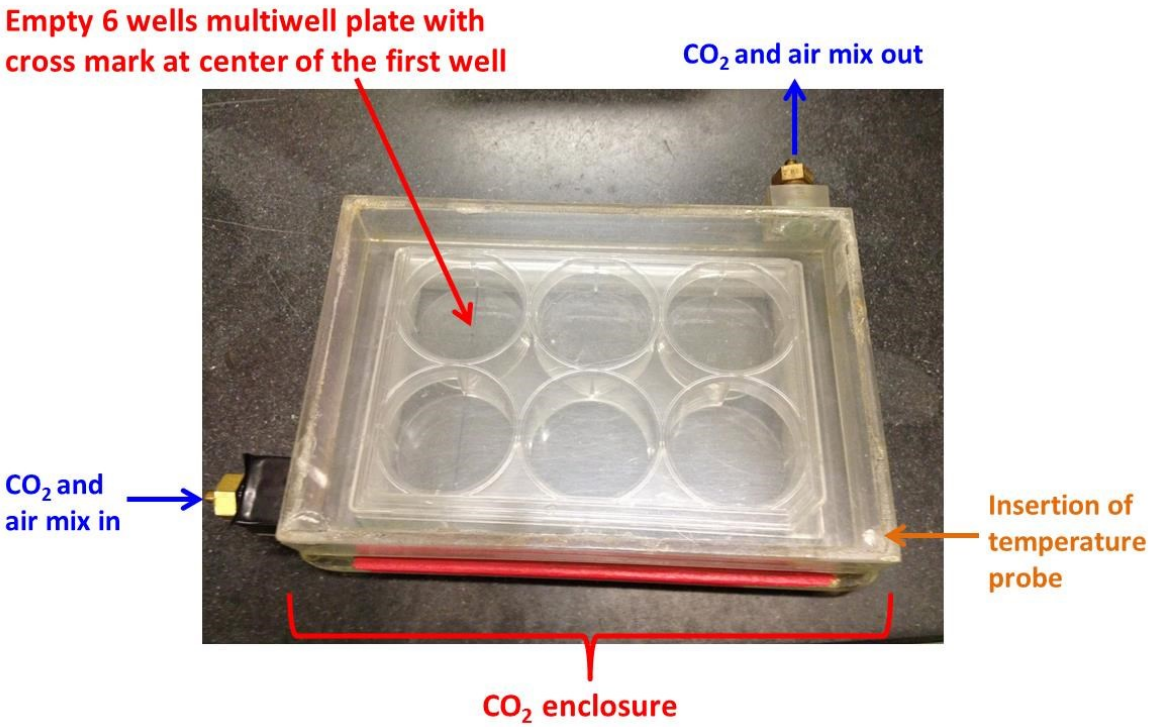


Figure 3-5 : CO_2 enclosure with ports for CO_2/air mix (secondary enclosure). This enclosure can be fitted on the automated stage.

3.3 Image acquisition

Images were acquired using Metamorph® from Olympus® (using the autofocus utility for each image). Every 3 hours, 121 images were captured in each well of 6 wells multiwell plates (using a grid of 11 images by 11 images, 0.4 mm apart). The number of fields of view was set empirically as to maximize as much as possible the area imaged in each well while maximizing the imaging frequency of each well (at least one acquisition every 3 hours). A total of 1,213,872 PCM images were used for data analysis in 40 experiments (240 wells), at a magnification of 20X. Images that were not of good quality (foggy, out of focus, etc.) were removed from the dataset. The complete list of experiments performed (with the experimental conditions) is provided in Appendix D.

Journals (.jnl) were built to acquire images at a given sampling time in several wells of a multiwell plate. The journals used lists of points (one list per well in a .STG file) generated by a Matlab® algorithm. The Metamorph® image acquisition procedure is provided in Appendix E. Using lists of points loaded by the MDA (Multiple Dimension Acquisition) function was deemed more efficient than using the Screen Acquisition function since it provided more flexibility for acquisition patterns.

3.4 Image processing and data analysis

Image processing and data analysis for this project were performed using Matlab® custom scripts, taking advantage of the Image Processing Toolbox®. The scripts created for this project were in part inspired from codes available from the Yawtb toolbox (Jacques, Coron et al. 2007), from the Medical Imaging Processing Toolbox® (Demirkaya, Asyali et al. 2009) and from CellProfiler® codes. Matlab® code lines provided by these sources were modified and changed significantly to meet the project requirements. Ground truth data was generated by manual segmentation of the individual cells using the Matlab® *roiploy* function. Cell features were then calculated from the segmented cells (e.g. minor and major axes, roundness and orientation) using the *bwlabeln* and the *regionprops* functions. The Matlab® script used to perform these operations (Metamorph_IMAGE_CALIBRATION_v0r2.m) is available in Appendix F.

For multivariate data analysis, SIMCA-P® and the PLS Toolbox® were used. To analyze the large dataset required for Chapter 6, resources from Calcul-Québec (COLOSSE) were used.

Matlab® scripts that were written for Chapter 4, Chapter 5 and Chapter 6 are provided in Appendix F.

Chapter 4. Selection and tuning of a fast and simple phase contrast microscopy image segmentation algorithm for measuring myoblasts growth kinetics in an automated fashion

Microscopy and Microanalysis journal, 19, 855–866, April 19 2013
doi:10.1017/S143192761300161X, reproduced with permission

Pierre-Marc Juneau[a], Alain Garnier[a], Carl Duchesne[a]*
[a]Department of Chemical Engineering, Pavillon Adrien-Pouliot, 1065 ave. de la Médecine, Université Laval, Québec (Québec), Canada, G1V 0A6
***Fax: +1 418-656-5993; Tel: +1 418-656-5184; E-mail:** *Carl.Duchesne@gch.ulaval.ca*

Short running title: Measuring myoblasts growth kinetics via PCM

Index terms: phase contrast microscopy, live-cell imaging, segmentation, cell-covered surface, degree of confluence, myoblasts, growth kinetics, passage number, algorithm tuning, algorithm selection.

4.1 Résumé

L'acquisition et le traitement des images de microscopie en contraste de phase dans le cadre d'applications d'imagerie en temps réel de cellules vivantes et de criblage à haut débit demeurent encore un défi puisque la méthodologie et les algorithmes utilisés doivent être rapides, simples à utiliser et à régler tout en étant le moins intrusifs possibles. Dans cet article, nous avons développé un algorithme simple et rapide pour calculer la surface recouverte par les cellules (degré de confluence) dans les images de microscopie en contraste de phase. Cet algorithme de segmentation est basé sur un filtre de plage de valeurs dont les paramètres sont : la taille de la plage , le seuil sur cette plage et le seuil de taille minimale des objets. Ces paramètres ont été ajustés de façon à maximiser la fonction *F-measure* sur un ensemble de 200 images de calibration segmentées à la main, et sa performance a été comparée à d'autres algorithmes proposés dans la littérature. Un ensemble comportant 1 million d'images provenant de 37 cultures de myoblastes cultivés dans des conditions différentes a été traité afin d'obtenir la surface recouverte par les cellules en fonction du temps. Les données ont été utilisées pour ajuster des modèles de croissance exponentiels et logistiques, et l'analyse a montré une relation linéaire entre les paramètres cinétiques et le nombre de passages tout en soulignant l'effet de la qualité du milieu de culture sur la cinétique de croissance des cellules. Cet algorithme peut être utilisé pour la surveillance en temps réel des cultures cellulaires et pour des expériences de criblage à haut débit après un réglage adéquat.

4.2 Abstract

Acquiring and processing phase contrast microscopy images in wide-field long-term live cell imaging and high throughput screening applications is still a challenge since the methodology and algorithms used must be fast, simple to use and to tune and as little intrusive as possible. In this paper, we developed a simple and fast algorithm to compute the cell-covered surface (degree of confluence) in phase contrast microscopy images. This segmentation algorithm is based on a range filter of a specified size, a minimum range threshold and a minimum object size threshold. These parameters were adjusted in order to maximize the F-measure function on a calibration set of 200 hand-segmented images, and its performance was compared to other algorithms proposed in the literature. A set of 1 million images from 37 myoblast cell cultures under different conditions were processed to obtain their cell-covered surface against time. The data was used to fit exponential and logistic models, and the analysis showed a linear relationship between the kinetic parameters and passage number and highlighted the effect of culture medium quality on cells growth kinetics. This algorithm could be used for real-time monitoring of cell cultures and for high throughput screening experiments upon adequate tuning.

4.3 Introduction

Wide-field, long-term live cell imaging (WFLT-LCI) is a very powerful technique which is rapidly spreading to study physiological and biological processes related to cell cultures such as cell state (morphology, quiescence), behavior (movement, growth, differentiation, death, interactions) and history using different imaging techniques (Goldman and Spector 2005; Rimon and Schuldiner 2011). In typical WFLT-LCI experiments, computer controlled incubator-stage positioning systems enable long time-series sequences of cell images in a number of wells to be acquired and analyzed using image processing techniques (Kenong, Gauthier et al. 1995; Moogk, Stewart et al. 2010; Leysi-Derilou, Duchesne et al. 2012). This enabling technology is now mature and can be key to important findings in a number of research fields such as wound healing, embryo development, immune system studies, cancer research, tissue engineering, stem cell research and drug discovery (Paduano, Sepe et al. 2010). WFTL-LCI is now frequently used in high throughput screening experiments, where large numbers of small cell samples are tested in parallel with minimal manual intervention (Freshney 2010), and for which a single experiment can generate hundreds of thousands of images with varying image quality (Kang, Miller et al. 2006). In order for WFLT-LCI to be successful for high throughput screening applications, it must meet at least three performance criteria: 1) the image acquisition must be fast and processing should require short computation times, 2) the image analysis algorithm must be robust and tuned properly so it is not affected by the large experimental variability inherent to cell culture and 3) as few chemical additives as possible should be used in order to avoid introducing artifacts in the biological experiments (Rimon and Schuldiner 2011).

Fluorescent dyes or markers are commonly used for live cell imaging (Rimon and Schuldiner 2011), which allow highlighting specific phenomena and facilitate tremendously image processing. Cells in an epifluorescence microscopy (EFM) image are easily segmented by using a simple intensity level threshold followed sometimes (to refine even more the results obtained) by the use of the conventional watershed algorithm on pixel intensities, gradients, shapes or a combination of all of those (Padfield, Rittscher et al. 2011). However, EFM live cell imaging experiments are usually of short duration, mainly due to the toxicity of the dyes and exposure to UV light (Frigault, Lacoste et al. 2009). It can therefore be advantageous to avoid adding these molecules to the culture and only use the microscope optics, such as in phase contrast microscopy (PCM) to enhance the image quality. Several authors have tackled the challenge of processing PCM images for

segmenting individual cells (Bradhurst, Boles et al. 2008; Debeir, Adanja et al. 2008; Liu, Li et al. 2011; Rapoport, Becker et al. 2011), which poses challenges for image processing such as uneven illumination and contrast enhancement (Rimon and Schuldiner 2011). Even though PCM images have the advantage of showing internal details (cell nucleus and granules) of cells in monolayers (Cox 2007), it can be difficult to identify individual cell borders accurately because of the bright and sometimes uneven halo surrounding them and to distinguish individual cells in a cluster because of blurry cell-cell interfaces (at high degree of confluence), broken borders and poor intensity contrast between the background and cellular regions (Bradbury and Wan 2010). More general difficulties inherent to WFTL-LCI such as uneven illumination (see Figure 4-3A), condensation on the plate cover, focus and positioning drift can also complicate the image processing step. Most image analysis software packages for WFTL-LCI platforms have been developed for processing primarily EFM images, and there is still a need for a fast and robust method that would be able to cope with the problems associated to PCM-WFTL-LCI image processing in high throughput screening experiments.

Methods for segmenting objects in images can be classified roughly into three categories: features classification, edge detection and region detection. When applied to PCM images, most of these methods require high computation times for performing single-cell segmentation, and have several disadvantages: features classification techniques such as intensity thresholding (Ridler and Calvard 1978; Otsu 1979; Kapur, Sahoo et al. 1985; Kittler and Illingworth 1986; Chowdhury and Little 1995) can be vulnerable to uneven illumination and noise (Adams and Bischof 1994), edge detection methods such as active contours (Kass, Witkin et al. 1988) and level set methods (Sethian and Adalsteinsson 1997) can struggle to reach the real contours of the cells when applied to PCM images (Orikawa and Tanaka 2010) and region detection methods such as the watershed transform (Vincent and Soille 1991) have a tendency to result into over-segmentation (Gonzalez and Woods 2008). Since the methods mentioned above for PCM-WFTL-LCI can be difficult to tune and to use while ensuring that the image analysis is robust, a solution is a compromise on the segmentation precision by discriminating cellular regions from background regions. This approach allows computing the degree of confluence (the portion of the surface covered by the cells) with sufficient accuracy to capture the speed of recolonization of an empty area (scratch wound approach, Debeir et al. (2008)) using PCM-WFTL-LCI images while testing different conditions. From a cell culture control perspective, the degree of confluence can be sufficient to estimate cell counts (Mõlder,

Sebesta et al. 2008; Topman, Sharabani-Yosef et al. 2011) or to determine the optimal subculture time (Ker, Weiss et al. 2011). Analyzing the degree of confluence against time allows the estimation of the cell population growth rate (He, Wang et al. 2007; Russell, Metaxas et al. 2007), assuming that the cell volume is proportional to the cell projected surface area and that they grow in monolayers *in vitro*, which is the mode of culture common to most normal adherent cells, with the exception of hematopoietic cells (Freshney 2010).

This paper presents an image processing method that is fast, simple and robust when properly tuned for segmenting cellular regions in PCM images, thus meeting all three performance criteria for using PCM-WFTL-LCI in high throughput screening applications. Experiments were performed on a primary cell line of human myoblasts over several passages using either a serum-supplemented (SSM) or a serum-free (SFM) culture medium, generating more than 1 million images over several months. The proposed algorithm was robust enough to process a large quantity of PCM images subject to uneven illumination conditions and different degrees of confluence. The three parameters of the image processing algorithm (range filter size, minimum range threshold value and minimum object size threshold value) were adjusted to optimize the accuracy of the segmentation using a set of manually-segmented images, and the performance of the algorithm was compared to other methods proposed in the literature. Kinetic models were fitted on the average cell-covered surface curves and statistical analysis showed that the kinetic parameters were close to values given in the literature and that they were linearly correlated with passage number ($N_{passage}$), showing the sensitivity and usefulness of the proposed method.

4.4 Materials and methods

4.4.1 Proposed segmentation algorithm

In order to segment cellular regions from background regions, the method proposed in this article consists of convoluting the image with a range filter to highlight regions of high signal amplitude and variance. During the convolution operation (Figure 4-1), the raw image (I) of size m×n is scanned with a square range filter window (RFW) of size RFS pixels×RFS pixels, generating a Range Map (RM) by computing the difference between the maximum and minimum values captured by the RFW for each of its individual positions in the raw image.

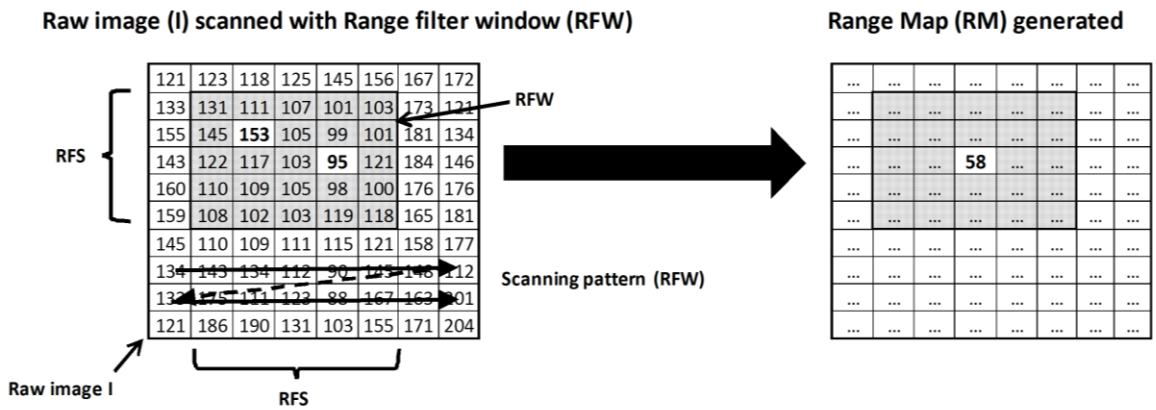


Figure 4-1: Generation of the Range Map (RM) with a range filter window (RFW) of size RFS (range filter size)

In PCM, regions occupied by cells usually have a signal of higher variance than background regions (Kenong, Gauthier et al. 1995) because of the microscope optics (Yin, Li et al. 2010), as can be seen in Figure 4-2A and B, where pixel intensity is plotted against position for a pixel line selected in a raw image. It is possible to locate cell-occupied regions in the image by looking at its RM (exemplified for the pixel line, Figure 4-2C). In order to highlight cellular regions, an intensity threshold is applied to the Range Map (RM) using a minimum range threshold value (MRT) to identify cellular and background regions, and generate a cellular region binary mask (CRBM) (equation 4-1):

$$\mathrm{CRBM}(\mathrm{i},\mathrm{j}) = \begin{matrix} 1 & \text{if } \mathrm{RM}(\mathrm{i},\mathrm{j}) => \mathrm{MRT} \\ 0 & \text{if } \mathrm{RM}(\mathrm{i},\mathrm{j}) < \mathrm{MRT} \end{matrix} \qquad (4\text{-}1)$$

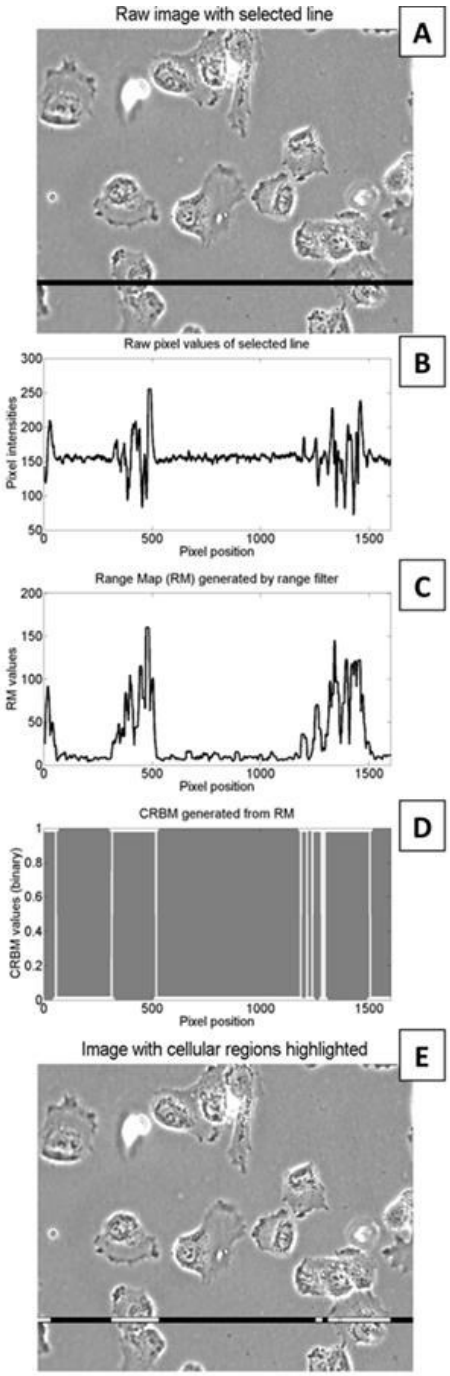


Figure 4-2: The segmentation method illustrated for a specific line of pixels in a phase-contrast microscopy image. A: Raw image and a selected line of pixels (black). B: Raw pixel values of a selected line. C: Range Map (RM) generated by a range filter. D: Cellular region binary mask (CRBM) calculated from RM. E: Image with cellular regions (white) highlighted.

The CRBM for the pixel line selected in the raw image is shown in Figure 4-2D, where the pixels associated with the cells are given a value of 1 while the background pixels are given value of 0 (white and black respectively). The selected pixels are shown in Figure 4-2E. The default *imfill* function in Matlab® is used on the CRBM matrix in order to fill the holes inside the pixel groups. To make sure that small artifacts in the image (e.g. cellular debris creating textured clusters of a few hundred pixels in size) are not falsely classified as cellular regions, pixel groups that are too small to be cells are removed. Each connected pixel group (P(k)) is highlighted (N_{pixel_groups} identified) and characterized using the default *bwlabeln* and *regionprops* functions in Matlab®. All the connected pixel groups, each of size (*size* P(k)), are compared to a minimum object size threshold (MOST, measured in pixels) and only those of size larger or equal to MOST are retained, forming the filtered cellular region binary mask (CRBMF) (equation 4-2):

for k=1 to N_{pixel_groups}

$$CRBMF(i,j) = \begin{matrix} 1 & \forall CRBM(i,j) \in P(k) \ \ such\ that\ size(P(k)) => MOST \\ 0 & \forall CRBM(i,j) \in P(k) \ \ such\ that\ size(P(k)) < MOST \end{matrix} \quad (4\text{-}2)$$

end

The cell-covered surface (S) is defined as the ratio of the number of pixels associated with cellular regions relative to the total number of pixels in an image (equation 4-3):

$$S = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} CRBMF(i,j)}{mn} \quad (4\text{-}3)$$

The proposed segmentation algorithm has therefore 3 parameters: the range filter size (RFS), which must be an odd number, the minimum range threshold value (MRT) and the minimum object size threshold (MOST). The output of this algorithm is the cell-covered surface (degree of confluence) also described by several authors (Athelogou, Eblenkamp et al. 2010; Ker, Weiss et al. 2011). The advantage of the proposed method is its simplicity (fast computation) and its ease of implementation and tuning, while providing a good accuracy in terms of precision and recall. The segmentation algorithms used in this article for imaging adherent cells are not able to distinguish viable from non-viable cells when computing the cell-covered surfaces. However, this has little effect on the results since dead cells usually detach from the plate surface and are not imaged.

4.4.2 Tuning of segmentation algorithms

The tuning (calibration) of a segmentation algorithm is a critical step in order to estimate optimal parameter values that will yield the best performance in different imaging conditions (i.e. robustness) and allow for a fair comparison of segmentation algorithms. However, the tuning of segmentation algorithms is seldom discussed in the literature (Al-Muhairi, Fleury et al. 2011). It usually requires a set of images segmented by human observers and referred to as “ground truth”. Most classification performance metrics rely on the computation of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Recall (R) is the ratio of TP over the sum of TP and FN, while precision (P) is the ratio of TP over the sum of FP and TP (Nattkemper, Twellmann et al. 2003; Dewan, Ahmad et al. 2011). Binary masks obtained by manual segmentation (M) and automatic segmentation (A) from images of size m by n are used to define R and P (He, Wang et al. 2007):

$$\mathrm{R} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{A}.*\mathrm{M})}{\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{M})} \tag{4-4}$$

$$\mathrm{P} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{A}.*\mathrm{M})}{\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{A})} \tag{4-5}$$

where the symbol ".*" denotes an element-by-element multiplication between two matrices of the same size. R and P are usually evolving in opposite directions, and it is required to use a weighted objective function to reach a compromise between both. The F-measure function (denoted by F) can be used for this purpose (Al-Muhairi, Fleury et al. 2011):

$$\mathrm{F} = \frac{\mathrm{P}\cdot\mathrm{R}}{\alpha\cdot\mathrm{R}+(1-\alpha)\cdot\mathrm{P}} \tag{4-6}$$

Where α is a weight parameter. Furthermore, we have also defined a second objective function that evaluates the relative error on the cell-covered area estimated (AEE):

$$\mathrm{AEE} = \frac{\mathrm{abs}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{M})-\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{A})\right)}{\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{M})} \tag{4-7}$$

The above metrics will be used for tuning and comparing the various segmentation algorithms tested in this work.

4.4.3 Kinetic growth models

The data associated to the cell-covered surface (S) as a function of time can be used to fit empirical growth models such as exponential or second order polynomial models (He, Wang et al. 2007; Ker, Weiss et al. 2011). Unbounded models consider that a cell population can grow without any limitations and bounded models consider that the cell population can't grow over a limited carrying capacity because of substrate or space limitation (Tabatabai, Williams et al. 2005). The most popular unbounded model is the exponential growth, which is expressed by a first order kinetic differential equation at low cell densities (Shuler and Kargi 1992):

$$\frac{dS}{dt} = \mu S \tag{4-8}$$

where μ is the constant specific growth rate. Among all models describing a bounded sigmoidal growth curve, one of the simplest is the logistic model (Vogels, Zoeckler et al. 1975). This model requires a maximum carrying capacity (S_{max}) and an initial cell-covered surface (S_0) to be solved, and is expressed by a non-linear differential equation (Shuler and Kargi 1992), where C_{max} is a growth constant (equation 4-9):

$$\frac{dS}{dt} = C_{max} S \left(1 - \frac{S}{S_{max}}\right) \tag{4-9}$$

4.4.4 Experimental set-up

Myoblasts from a primary cell line (kindly provided by Dr. J.P. Tremblay, CRCHUL) were cultivated over 9 passages in SSM (MCDB 120 + 15% FBS) and in SFM (Parent 2009) in 6-well plates. Both media used MCDB120, which is an optimized nutrient medium for human muscle stem cells (Ham and Blau 1988). Target inoculation density was in the range of 5,000-20,000 cells/mL, with a working volume of 2 mL/well. Growth medium was changed every 2-3 days. Phase contrast images were captured with an IX-81 microscope (Olympus™) equipped with an automated stage and a Qimaging® monochrome camera at 20X. An incubator was mounted on the microscope stage to maintain the culture plate at 5% CO_2 and 37.5 ˚C. The experimental set-up was controlled using Metamorph® (Olympus™), and the focus was adjusted for each frame using the autofocus utility. Every 3 hours, 121 images per well were captured, following an 11 x 11 fields of view grid, 0.4 mm apart. A total of 1,058,145 images were acquired over 37 experiments.

4.5 Results

4.5.1 Selecting the objective function

A calibration set of 200 ground truth images selected to span the whole range of degrees of confluence and conditions (roughly one image per experiment per well, at different conditions, time points and confluence) was used to optimize the segmentation model parameters. These images were manually segmented by drawing the cell contours using the *roipoly* function in Matlab® to create segmentation binary masks.

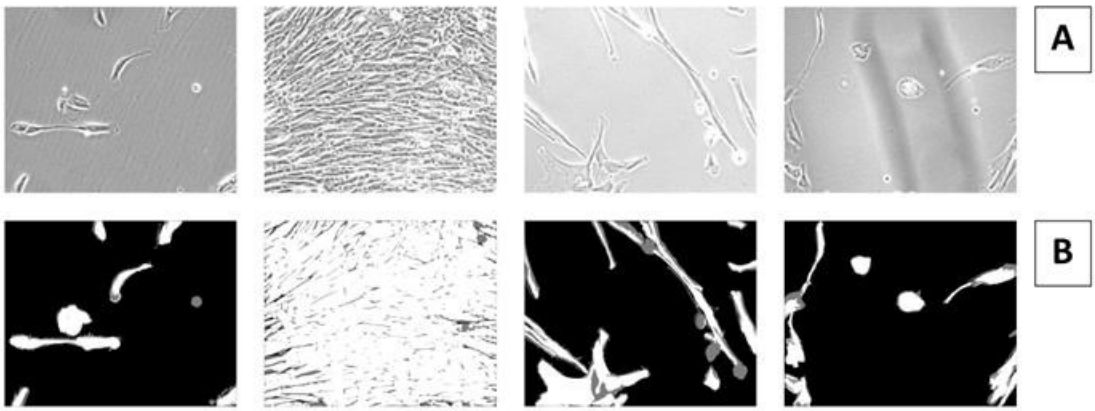


Figure 4-3: Examples of ground truth images used. A: Raw images. B: Result of the optimized segmentation algorithm. White is true positive, grey is false positive/false negative, and black is true negative.

Figure 4-3A shows a few examples of ground truth images (representative of the whole dataset in terms of variety) and Figure 4-3B shows the results of the segmentation algorithm applied to those images after optimization. To maximize the F function (for each value of α) or to minimize the AEE, a grid search was conducted by varying the RFS from 1 to 35 (by increments of 2), the MRT from 0 to 100 (by increments of 5) and the MOST from 0 to 10,000 (by increments of 1,000). When α=0, F=R and maximizing F tends to select the whole image in order to optimize recall. When α=1, F=P and maximizing F tends to reject a lot of pixels in order to keep only pixels that are the most certainly associated to cells (to optimize precision). For intermediate values of α, maximizing the F function tends to reach a compromise between R and P, giving more or less importance to each variable depending of the value of α. For each set of parameters, the average R, P, F and AEE were computed for the 200 images. Different values of α (from 0 to 1) were set to assess how the compromise (reflected in the F function) affected the segmentation performance.

Figure 4-4 illustrates the performance of the segmentation algorithm on a particular ground truth image and Table 4-1 shows the results obtained for different optimization criteria (approximate confidence intervals were calculated from the standard deviation observed in all the ground truth images). The results showed that P is positively correlated to MRT and that R is positively correlated to RFS and MOST. In fact, increasing MRT means that more pixels are rejected (being more selective means a higher P) and increasing RFS and MOST tends to classify more pixels as belonging to cellular regions (higher R).

Table 4-1 illustrates that maximizing the F function for α=0.5 provides a good compromise between R (87.7%) and P (77.4%) while having a good performance for the AEE (26.3%). For α=0.5, the optimal parameter values are: RFS=3, MRT=10 and MOST=3000. Maximizing F is preferred over minimizing AEE since the absolute surface error doesn't take into account the spatial performance of the algorithm, which is supposed to decrease both FP and FN.

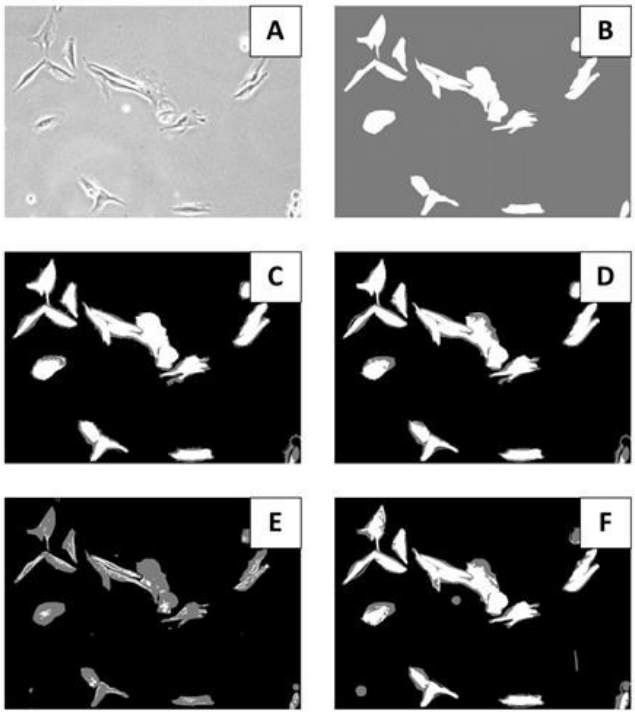


Figure 4-4: Example of the segmentation algorithm performance for different optimization criteria: (A) original image; (B)-(E) maximization of the F function for (B): α=0, (C): α=0.5, (D): α=0.7 and (E): α=1; (F) minimizing the area estimated error (AEE). White pixels represent true positive, grey pixels represent false positive/false negative, and black pixels represent true negative.

Table 4-1: Segmentation parameters obtained by optimization

Criteria of optimization	α	RFS	MRT	MOST	Average R	Average P	Average F	Average AEE
Maximizing F	0	35	0	0	1.000 ± 0.002	0.212 ± 0.221	1.000	9.039
	0.1	9	20	2000	0.965 ± 0.087	0.630 ± 0.172	0.916	0.983
	0.2	11	25	3000	0.949 ± 0.102	0.668 ± 0.141	0.875	0.645
	0.3	7	20	3000	0.923 ± 0.121	0.723 ± 0.122	0.852	0.415
	0.4	3	10	3000	0.877 ± 0.148	0.774 ± 0.099	0.833	0.263
	0.5	3	10	3000	0.877 ± 0.148	0.774 ± 0.099	0.822	0.263
	0.6	3	10	4000	0.868 ± 0.153	0.779 ± 0.096	0.812	0.251
	0.7	5	20	3000	0.820 ± 0.180	0.804 ± 0.070	0.809	0.207
	0.8	5	20	3000	0.820 ± 0.180	0.804 ± 0.070	0.807	0.207
	0.9	7	30	3000	0.780 ± 0.191	0.812 ± 0.067	0.809	0.202
	1	7	100	0	0.187 ± 0.131	0.870 ± 0.120	0.870	0.788
Minimizing AEE	-	9	40	1000	0.768 ± 0.184	0.791 ± 0.078	-	0.199

4.5.2 Comparative performance of the segmentation algorithms

To confirm the choice of the proposed algorithm over other methods found in the literature (for this particular application), several algorithms were tuned on the same ground truth images dataset by grid search while maximizing F for α=0.5. Table 4-2 describes the parameters and the grid search boundaries and steps used to tune each algorithm studied and Table 4-3 shows the optimum parameters and performance results for each algorithm. The proposed method (1) was first compared to some of its variants using a standard

deviation filter (2) or an entropy filter (3) of size FS followed by thresholding (MT) and by filtering out pixel groups smaller than MOST. Method 4 is based on the algorithm proposed by Topman et al. (Topman, Sharabani-Yosef et al. 2011) using the intersection of thresholded standard deviation maps (with two windows of different sizes, bW and sW, thresholded using MT) followed by morphological operations. In method 5, as part of a more advanced algorithm for detecting cells nuclei and performing cell tracking, the approach proposed by Dewan et al. (Dewan, Ahmad et al. 2011) uses a top-hat filter on the inverted image with a circular non flat structuring element of radius r and height h, followed by Otsu's thresholding method. In method 6, the approach proposed by Huth et al. (Huth, Buchholz et al. 2010) for coarse cell region segmentation is the multiplication of the output of an entropy filter with the output of a median filter (sizes EFS and MFS respectively) followed by Otsu's thresholding (Otsu 1979) and completed by filtering out pixel groups smaller than a certain number of pixels (MOST). Method 7 is an adapted version of the algorithm for rough cell-covered area segmentation proposed by Bradhurst et al. (Bradhurst, Boles et al. 2008), in which the image is filtered with a standard deviation filter of size FS, thresholded using the minimum error thresholding technique (Kittler and Illingworth 1986) and cleaned by filtering pixel groups smaller than MOST and holes smaller than MHST. Finally, in method 8, for detecting the cell-covered area in each image, a soft morphological gradient filter of size FS (subtracting the 20% percentile grey level value from the 80% percentile grey level value captured by the filter) followed by a threshold of value MT was proposed by Debeir et al. (Debeir, Megalizzi et al. 2008).

Table 4-2: Algorithms parameters and grid search settings

Algorithms	Parameters	Low limit	Step	High limit
1- Range filter	A) FS	1	2	35
	B) MT	0	5	100
	C) MOST	0	1000	10000
2- Standard deviation filter	A) FS	1	2	35
	B) MT	0	2	40
	C) MOST	0	1000	10000
3 - Entropy filter	A) FS	1	2	15
	B) MT	0	1	20
	C) MOST	0	1000	10000
4 - Topman's algorithm	A) bW	1	2	35
	B) sW	1	2	35
	C) MT	0	0.01	0.30
5 – Dewan's approach	A) r	1	2	75
	B) h	1	2	55
6 – Huth's approach	A) EFS	1	2	35
	B) MFS	1	2	55
	C) MOST	0	500	10000
7 – Bradhurst's approach	A) FS	1	2	35
	B) MOST	0	500	10000
	C) MHST	0	500	15000
8 – Debeir's approach	A) FS	1	2	35
	B) MT	1	2	150

Table 4-3: Performance comparison between algorithms on ground truth images

Algorithms	Parameters			Performance criteria				
	A	B	C	R	P	F	AEE	ACT (s)
1- Range filter	3	10	3000	0.88 ± 0.15	0.77 ± 0.10	0.81 ± 0.11	0.26± 0.32	0.31 ± 0.16
2- Standard deviation filter	7	6	3000	0.87 ± 0.16	0.77 ± 0.09	0.80 ± 0.11	0.24 ± 0.19	0.67 ± 0.06
3- Entropy filter	7	4	4000	0.86 ± 0.15	0.76 ± 0.10	0.79 ± 0.11	0.27 ± 0.32	2.89 ± 0.36
4- Topman's algorithm	29	1	0.05	0.86 ± 0.13	0.72 ± 0.10	0.78 ± 0.10	0.26 ± 0.22	0.27 ± 0.04
5 – Dewan's approach	67	53	-	0.51 ± 0.08	0.78 ± 0.11	0.61 ± 0.08	0.35 ± 0.13	3.28 ± 0.08
6 – Huth's approach	25	53	8000	0.85 ± 0.08	0.69 ± 0.10	0.75 ± 0.07	0.29 ± 0.16	14.21 ± 1.64
7 – Bradhurst's approach	7	3500	10000	0.89 ± 0.20	0.77 ± 0.08	0.80 ± 0.17	0.30 ± 0.21	0.48 ± 0.02
8 – Debeir's approach	27	19	-	0.86 ± 0.12	0.68 ± 0.12	0.75 ± 0.10	0.36 ± 0.34	0.67 ± 0.05

These algorithms were tested and compared on the ground truth image dataset while using the optimum parameters for each. It is interesting to see that most of the proposed algorithms provided have a similar performance when properly tuned. However, the results showed that the range filter algorithm provides a slightly better performance in terms of R and P than the other algorithms while having a competitive average computing time per image (ACT), thus justifying its selection. From a robustness perspective, method 6 provides a slightly better performance in comparison with the other algorithms by providing a smaller standard deviation on the average objective function F, but at the expense of a lower recognition performance (lower F) and a much higher computation time (ACT). Note that for all the proposed methods, no background correction or pre-processing was used. These tests were performed on an Intel® Core™ i7-2630QM CPU @ 2.00 GHz computer with 4Go of RAM.

The wide variety of segmentation algorithms, imaging problems, and applications available in the literature confirm that image processing and analysis is problem dependent and that no single method will work best for all PCM-WFTL-LCI applications. Therefore, from a practical perspective, several alternative segmentation algorithms should be compared on a set of ground truth images representing the range of conditions expected in practice. A

very important point to ensure a fair comparison between the algorithms is to tune their parameters based on the same objective function based on segmentation accuracy, robustness, computing time or other relevant performance metrics. Then, the most suitable algorithm can be selected for the particular application.

4.6 Discussion

4.6.1 Application 1: cell-covered surface (S) growth curves

The cell-covered surface (S) was calculated for each of the 1.05 million images with the optimized range filter segmentation algorithm (looking at each individual well in the 37 experiments, for a total of 222 growth curves).

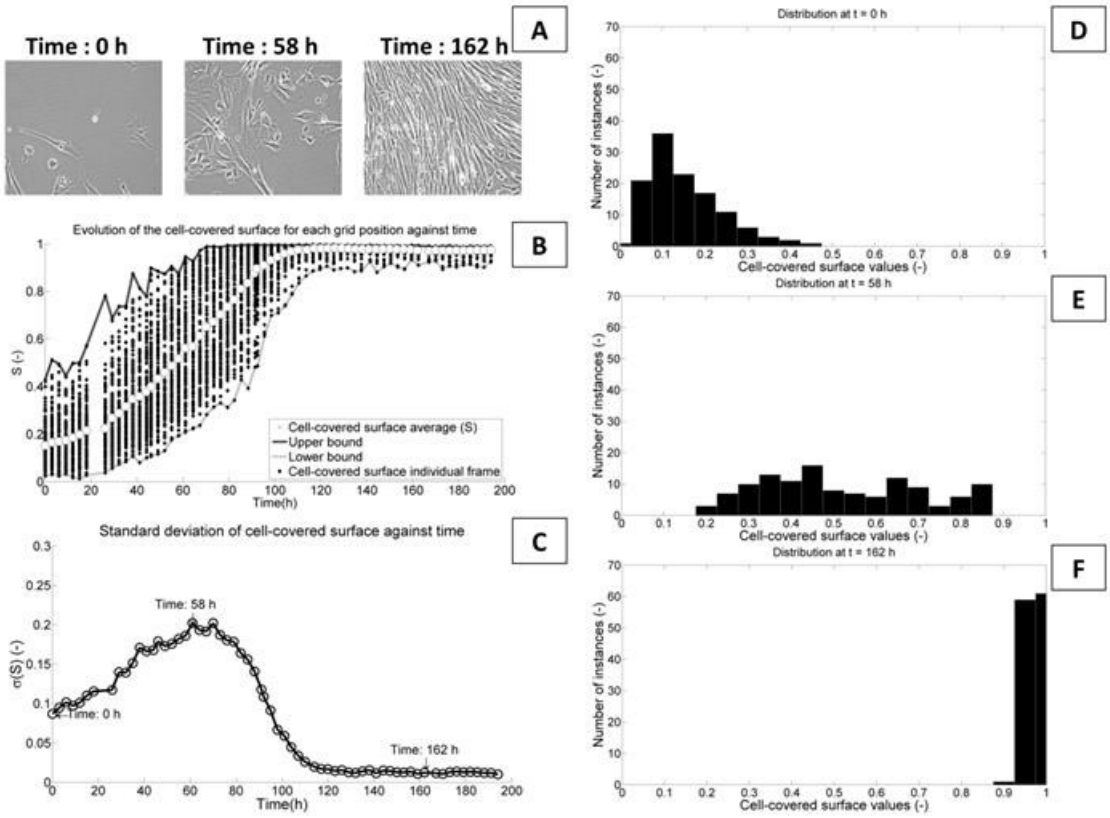


Figure 4-5: A: Typical images of the evolution of a culture (from low to high cell-covered area); B: Evolution of S against time for each position in the acquisition grid, and average cell-covered surface (white dots) as a function of time; C: Standard deviation of S against culture time; Distribution of the individual cell-covered surfaces at (D): 0h, (E): 58h, and (F): 162h.

Figure 4-5A shows how the degree of confluence increases through time for images from the same sequence, and Figure 4-5B shows the evolution of the cell-covered surface against time for each position in a well (121 frames per acquisition) for one experiment. The lower bound is associated with the minimum cell-covered surface for each acquisition and the upper bound corresponds to the maximum value. The average cell-covered surface ($\bar{S}_t$) and the standard deviation ($\sigma(S_t)$) at each acquisition time is computed from individual cell-covered surfaces (S(f,t)):

$$\bar{S}_t = \frac{\sum_{f=1}^{121} S(f,t)}{121} \tag{4-10}$$

$$\sigma(S_t) = \sqrt{\frac{\sum_{f=1}^{121}(S(f,t)-\bar{S}_t)^2}{121}} \tag{4-11}$$

Figure 4-5B shows that the average cell-covered surface (for the whole acquisition grid, at each point in time, illustrated by white dots) can provide an overall measure of the degree of confluence in each well. Figure 4-5C shows the spatial variance $\sigma(S_t)$ of the cell-covered surfaces for each acquisition through time. Figure 4-5D/E/F shows that at the beginning of a cell culture, the distribution of S follows a Poisson distribution as the cells spread randomly over the surface (normally associated with hemacytometer counts (Nielsen, Smyth et al. 1991)) and transitions slowly to a uniform distribution (Walck 1996) as the culture goes on and cells occupy the entire field of view. For each well, the curve of the average cell-covered surface ($\bar{S}_t$) as a function of time was fitted with the exponential model, varying the number of points from 5 to 50, and keeping the number of points maximizing the model coefficient of regression, R^2. The logistic model was fitted on the entire curves. The differential equations were solved using the 4th order Runge-Kutta method (Kreyszig 2000) and the models were fitted using non-linear regression with Matlab®.

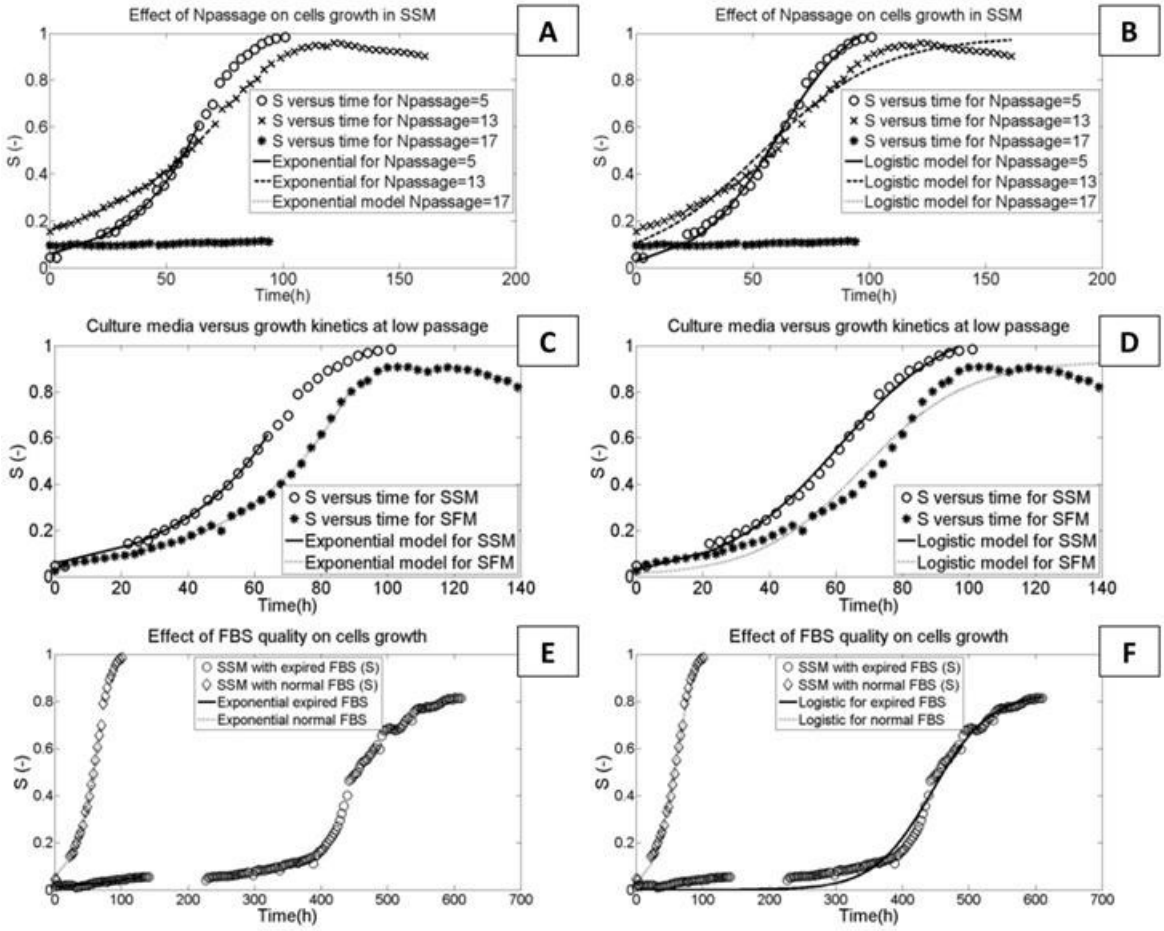


Figure 4-6: Fitting of the kinetic models on the average cell-covered surface. Effect of passage number on growth kinetics of cells grown in serum-supplemented medium (SSM) for different kinetic models (A) Exponential; (B) Logistic. Comparison between SSM and serum-free medium (SFM) at $N_{passage}$=6 for different kinetic models (C) Exponential; (D) Logistic. Effect of FBS quality on cells growth kinetics for different kinetic models (E) Exponential; (F) Logistic.

Figure 4-6A/B shows the impact of the number of passages ($N_{Passage}$) on the cell-covered surface curves for cells grown in SSM. For $N_{Passage}$ = 5, a μ of 0.87 day^{-1} and a C_{max} of 1.39 day^{-1} were calculated. For $N_{Passage}$ = 13, a μ of 0.45 day^{-1} and a C_{max} of 0.94 day^{-1} were computed, showing a slower growth in comparison to lower passage cells. For $N_{Passage}$ = 17, a μ of 0.05 day^{-1} and a C_{max} of 0.06 day^{-1} were calculated, showing practically no growth at high passage. According to Parent (2009), myoblasts in either SSM or SFM have similar growth kinetics, and it is interesting to compare the average cell-covered surface curves for both media for the same passage. Figure 4-6C/D shows two typical cell-covered

surface curves for both the SSM and SFM at $N_{Passage}$ = 6. Fitting the average cell-covered surface with the models, we computed a μ of 0.87 day^{-1} and a C_{max} of 1.39 day^{-1} for the SSM and a μ of 0.78 day^{-1} and a C_{max} of 1.52 day^{-1} for the SFM, which are not significantly different. Analyzing the cell-covered surface curves can help troubleshooting cell culture problems for process and quality control purposes. For example, Figure 4-6E/F shows the cell-covered surface growth curves for low passage myoblasts grown in SSM with expired and normal FBS. Fitting the average cell-covered surface with the models, we computed a μ of 0.87 day^{-1} and a C_{max} of 1.39 day^{-1} for the normal FBS compared to a μ of 0.25 day^{-1} and a C_{max} of 0.55 day^{-1} for the expired FBS.

Finally, note that Mõlder et al. (2008) and Topman et al. (2011) have shown that a linear relationship exists between the cell-covered surface (S) and cell count. We also confirm that a similar trend exists (R^2=0.80) for the cellular system studied in this work (results not shown).

4.6.2 Application 2: Correlation between kinetic parameters and culture conditions (passage number)

For each well for all the 37 experiments, the average cell-covered surface was fitted with the exponential and logistic models and kinetic parameters were compiled. Since image quality, experimental noise and initial values for the non-linear regressions can affect the goodness of fit, only the parameters meeting specific quality criteria were considered before studying their relationship with $N_{passage}$. For the exponential and the logistic models, only the kinetic parameters statistically significant up to a confidence interval of 95% (under a Student-t test) and associated with a model that had a R^2 higher than 95% were considered. For the exponential model, in addition to those two conditions, only the kinetic parameters calculated with an initial cell-covered surface (S_0) lower than 50% were kept, in order to ensure that surface availability wouldn't impact the analysis.

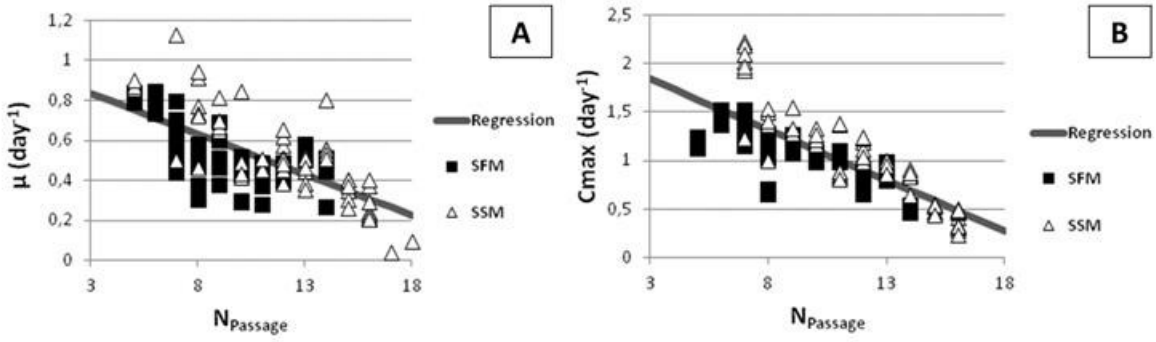


Figure 4-7: Kinetic parameters for each model as a function of $N_{Passage}$ meeting all criteria. (A) μ for the exponential model; (B) C_{max} for the logistic model.

Figure 4-7 shows the relationship between the kinetic parameters considered (meeting all criteria) and $N_{passage}$. To characterize the impact of passage number ($N_{Passage}$) on each kinetic parameter ($K(N_{Passage})$), a linear relationship was assumed:

$$K(N_{Passage}) = aN_{Passage} + b \quad (4\text{-}12)$$

Table 4-4 provides the linear relationship parameters between $N_{Passage}$ (passages 5 to 18) and the kinetic parameters. The intervals over the *a* and *b* parameters were calculated using a Student-t test with a confidence interval of 95%. The kinetic parameters obtained are consistent with the fact that as the myoblasts multiply through passages ($N_{Passage}$ increasing), their growth rate decreases until it reaches zero (no growth) around the 18th passage. For the exponential model, at low passages, the specific growth rates (μ) obtained are within the range of values provided in the literature via manual cell counts (0.57 to 0.78 day^{-1}, Boudreault et al.(2001)) and by automatic image processing (0.64 day^{-1} for healthy cells and 0.37 day^{-1} for starving cells, He et al.(2007)), meaning that the exponential model parameters obtained are close to independent observed values. From a fresh biopsy, a cell cycle time as low as 0.67 days (μ of 1.03 day^{-1}) was reported (Deasy, Jankowski et al. 2003), a value close to the *b* parameter of 0.96 day^{-1} that would be achieved at $N_{passage}$=0. In addition, the exponential model can show adequately the impact of $N_{passage}$ on cells kinetics with a linear correlation having a R^2 of 51.0%. The logistic model seems to be more robust, since the C_{max} parameter correlates even better with $N_{Passage}$ for both media (R^2 of 66.1%).

Table 4-4: Linear relationship parameters between kinetic parameters and passage number (for $N_{Passage}$ 5 to 18)

Models	Parameters	a	b	R^2
Exponential	μ (day^{-1})	-0.0407 ± 0.0066	0.958 ± 0.074	0.5098
Logistic	C_{max} (day^{-1})	-0.1041 ± 0.0134	2.151 ± 0.151	0.6614

4.7 Conclusion

In order to use PCM-WFTL-LCI for high throughput screening applications, the image acquisition and processing methods must meet three criteria: they must be fast, tuned properly so it is robust to image variations and non-intrusive. This paper presents a simple method that can be used to process PCM images for this purpose. A simple, efficient and easily tuned segmentation algorithm based on a range filter was successfully used and optimized to segment cellular regions from background with a slightly higher accuracy compared to other algorithms reported in the literature. The segmentation algorithm was used on the whole PCM images database (1,058,145 images acquired over 37 experiments, generating 222 growth curves) to compute the average cell-covered surface against time for each well, and the exponential and logistic models were fitted on the data. The two models showed clearly a linear relationship between $N_{Passage}$ (age of the cells) and the kinetic parameters, thus illustrating the great potential of the approach.

The proposed method can be used for online cell-culture monitoring to determine the moment at which cells are confluent when producing cells for tissue engineering purposes (e.g. skin grafts) and also to determine when to harvest cells for cellular therapy applications (e.g. healthy myoblasts for transplantation in dystrophic patients). For tissue engineering applications, the proposed method has the advantage of being non-intrusive, which is essential in order to be able to transplant the healthy cells to patients (without the use of harmful additives such as fluorescent dyes). In HT screening experiments, the main advantage is to be able to assess the growth kinetics of adherent cells without having to sacrifice cultures (by trypsinization) and without using dyes that could bias cellular behavior. For example, the proposed method can be used to assess the growth kinetics of myoblasts while screening different serum-replacement growth factors.

4.8 Ackowledgements

The authors would like to acknowledge the lab of Dr. J.P. Tremblay (CRCHUL) for providing expertise/material for the culture of myoblasts, and the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) for funding.

Chapter 5. The Undecimated Wavelet Transform - Multivariate Image Analysis (UWT-MIA) for simultaneous extraction of spectral and spatial information

Chemometrics and Intelligent Laboratory Systems, 142, 304-318, March 2015
doi: 10.1016/j.chemolab.2014.09.007, reproduced with permission

Pierre-Marc Juneau[a], Alain Garnier[a], Carl Duchesne[a]*
[a]Department of Chemical Engineering, Pavillon Adrien-Pouliot, 1065 ave. de la Médecine, Université Laval, Québec (Québec), Canada, G1V 0A6
***Fax: +1 418-656-5993; Tel: +1 418-656-5184; E-mail:** *Carl.Duchesne@gch.ulaval.ca*

Short running title: Undecimated Wavelet Transform Multivariate Image Analysis (UWT-MIA)

Index terms: Undecimated Wavelet Transform, Gabor, Multivariate image analysis, Texture analysis, Principal component analysis, Segmentation

5.1 Résumé

Combiner l'information associée à la couleur (spectrale) et à la texture conduit souvent à une amélioration des performances des capteurs d'imagerie développés pour différentes applications, comme la surveillance des procédés et la prédiction des variables de production ou de qualité du produit. Une nouvelle méthode basée sur la transformée en ondelettes non décimée et l'analyse multivariée des images (UWT-MIA) est proposée dans cet article pour l'extraction simultanée de l'information spatiale et spectrale. Les principaux avantages de cette approche sont: 1) elle utilise des ondelettes continues qui ont une résolution constante sur toutes les échelles par rapport aux ondelettes discrètes, et sont mieux adaptées pour extraire des caractéristiques de distribution de taille, 2) les images de détails et d'approximation des ondelettes ont la même taille et sont spatialement congruentes. Ceci permet l'empilement des sous-images obtenues à toutes les échelles, orientations et pour toutes les longueurs d'onde pour les images multi- et hyperspectrales. Ainsi, chaque sous-image est considérée comme un canal séparé dans une nouvelle image multivariée à partir de laquelle un seul modèle MIA peut être construit. La performance de la méthode UWT-MIA est illustrée à l'aide d'images synthétiques et naturelles. Il est démontré que les variations de couleur, de taille et d'orientation des objets d'intérêt peuvent être suivies efficacement dans le même modèle MIA. La méthode proposée peut également être intégrée dans le cadre des méthodes d'analyse d'image multivariée multirésolution (MR-MIA) proposées dans la littérature.

5.2 Abstract

Combining color (spectral) and textural information often leads to improved performance of image-based sensors developed for different applications, such as process monitoring and prediction of process or product quality variables. A new method based on the Undecimated Wavelet Transform and Multivariate Image Analysis (UWT-MIA) is proposed in this paper for simultaneous extraction of spatial and spectral information. The key advantages of this approach are: 1) it uses continuous wavelets which have a constant resolution over all scales compared to discrete wavelets, and are better suited for extracting size distribution features, 2) the wavelet detail and approximation sub-images have the same size and are spatially congruent which allows stacking the sub-images obtained at all scales and orientations (and wavelengths for multi- and hyperspectral images) as a separate channel in a new multivariate image from which a single MIA model can be built. The performance of UWT-MIA is illustrated using both synthetic and natural images. It is shown that variations in color, size and orientation of objects of interest can be tracked efficiently within the same MIA model. The method can also be cast within the multiresolutional multivariate image analysis (MR-MIA) framework proposed in the literature.

5.3 Introduction

In the last decade, several machine vision approaches based on spectral and/or textural analysis have been developed for process and product quality monitoring and control, many relying on Multivariate Image Analysis (MIA) and Multivariate Image Regression (MIR) (Prats-Montalban, de Juan et al. 2011; Duchesne, Liu et al. 2012). An important advantage of MIA-based methods is that image segmentation is performed within the multivariate feature space (i.e. masks in the score space) (Geladi and Grahn 1996) rather than directly in the image space as with traditional machine vision techniques (Gonzalez and Woods 2008). Hence, in a monitoring context, the same masks are used to segment new incoming images as long as the MIA model remains valid. Drawing of new boundaries (masks) is performed only when deemed necessary because e.g. the model requires updating. Traditional image segmentation techniques calculate new boundaries for every new incoming image. Color/spectral segmentation using MIA/MIR was used to develop very efficient image-based sensors for the process industry (Duchesne, Liu et al. 2012). More recently, MIA was considered for texture analysis (Lied, Geladi et al. 2000; Bharati, Liu et al. 2004), but ways to reintroduce spatial information in MIA were investigated since image pixels lose their spatial identity during the unfolding operation. The basic idea that was put forward in the literature consists of computing several filtered versions of the same grey level image, stacking them as different channels of a multivariate image, and then applying standard MIA. The main difference between the approaches proposed in the literature is the method used to filter the images.

Texture filters traditionally used in the image processing field (Gonzalez and Woods 2008) were among the first to be proposed for reintroducing spatial information in MIA. These filters are typically expressed as some linear combinations of pixel intensities (e.g. discretized derivative filters) or consist of local statistics (e.g. median filter) computed using intensities of neighboring pixels. Lied et al. (2000) proposed the multivariate image texture analysis method (MIX) in which a series of textural image derivatives generated by different filters (median, Laplace, compound) are stacked as additional spectral variables prior to performing MIA. The image shifting and stacking approach introduced by Bharati et al. (2004) creates a multivariate image by shifting a grey level image by one or more pixels in 8 directions and stacking the shifted images after appropriate cropping. MIA is then applied on the new multivariate image. In the latter method, the PCA loadings of each neighboring pixel define a set of spatial filters (i.e. one per PCA component) completely determined by the data itself. Prats-Montalbán and Ferrer (2007) extended the use of the

shifting and stacking approach to color-texture (spatial-spectral) analysis by applying it on each color channel separately and augmenting the texture matrix. The computation cost associated to the shifting and stacking method was also reduced by using the Fast Fourier Transform and the convolution theorem (improved MIA or iMIA) to look at pixel correlations in larger neighborhoods (Facco, Masiero et al. 2011).

Transform-based filtering techniques were also proposed to create multiple filtered versions of a grey level image in the spectral or spatial-spectral domains (i.e. filtering is based on the image frequency content). Geladi (1992) used the 2D Fast Fourier Transform (FFT) to provide a common basis for the analysis of a series of images of different sizes. The stack was formed by the magnitude images at each frequency (i.e. so-called the magnitude spectrum). The method was applied to compare textural patterns within two series of images (wood chips and botanical residues). More recently, Liu and MacGregor (2007) proposed the Multiresolutional Multivariate Image Analysis (MR-MIA) framework for spatial-spectral image analysis based on the 2D Wavelet Texture Analysis (WTA). Two methods were proposed. MR-MIA I simultaneously extracts spatial-spectral information by applying WTA to each spectral channel first and then by stacking the detail sub-images (wavelet planes) of each spectral channel separately for each scale (s =1,2,…S), producing sub-images for each orientation that are congruent for a given scale (can be stacked). MR-MIA II extracts spatial-spectral information separately by applying WTA to the score images computed using MIA as the first step. The choice of using either MR-MIA I or II depends upon whether the spatial-spectral information is interacting (e.g. as in hyperspectral images) or not.

Even though the proposed methods provide solutions for reintroducing spatial information in MIA, they also have some limitations. For instance, the methods using derivative filters, including the shifting and stacking approach, were found less powerful and flexible than transform-based methods, which are able to selectively remove noise and low frequency variations (e.g. caused by uneven lighting conditions) (Bharati, Liu et al. 2004). In the case of the MIX method, filter selection is not straightforward and requires some trial-and-error to determine the optimal set of filters to use for a given application. Applying MIA to a stack of magnitude images obtained using the 2D FFT allows discriminating images having different frequency content but the spatial location of the features of interest is lost because the FFT itself has a good frequency resolution but a low spatial resolution . MR-MIA methods solve this issue by using wavelets, which have both a good frequency and

spatial resolution (Gonzalez and Woods 2008) (compared to the FFT). However, the MR-MIA methods rely on the discrete wavelet transform (DWT (Mallat 1989)), which, in general, yield wavelet planes (i.e. wavelet sub-images) that are not congruent across different scales due to dyadic image sub-sampling. Therefore, in the MR-MIA I method, applying DWT as the first step yields as many new multivariate images as the number of scales (S) which are then analyzed separately using MIA. As discussed in Liu and MacGregor (2007), it is optimal to analyze these new multivariate images separately because DWT uses an orthogonal wavelet basis, meaning that the information contained in each one of the new multivariate images is independent from the others. The orthogonality of the wavelet planes at different scales proved very useful for applications such as image compression (Starck, Elad et al. 2004). However, in some image analysis applications, it may be interesting/necessary to combine the features extracted at different scales and orientations in a single model, for instance to track variations in the size distribution of certain objects of interest in a time series of images (set of images). In that case, the MR-MIA approach would require collecting the features extracted at different scales in a row vector for each image of the set followed by building a PCA model for comparing the images between them. These additional modelling efforts may be avoided by using the Undecimated Wavelet Transform (UWT) instead of the DWT, since the UWT provides spatially congruent wavelet planes. Hence, the UWT would allow combining the information extracted at different scales and orientations for different color channels or wavelengths (i.e. if applied to a multivariate image) using a single MIA model.

This paper presents the Undecimated Wavelet Transform Multivariate Image Analysis (UWT-MIA), a new transform-based method for spatial-spectral analysis of multivariate images, but also applicable to grey level images. The advantages of UWT-MIA over the discrete wavelet texture analysis (DWT) used in MR-MIA are two-fold: 1) UWT-MIA uses the continuous Gabor wavelet transform (CWT) for higher frequency and spatial resolutions (yields a constant resolution over all scales, which is an advantage for estimating size distributions as shown in Liu and Macgregor (2008)) and 2) the wavelet planes (matrices of wavelet coefficients) are of the same size and spatially congruent. Hence, both textural and spectral patterns could be segmented at the pixel level in the score space of a single MIA model. The main objective of the paper is to demonstrate the latter. The proposed method can also be seen as an extension of MR-MIA methods, combining the information extracted at different scales, orientations and wavelengths for each pixel of the image in a single stack on which MIA is performed. Even if the UWT-MIA

methodology presented in this article is suitable for simultaneous extraction of spatial and spectral information (i.e.as MR-MIA I), it could also be implemented within a MR-MIA II framework (separate extraction of spectral and textural features).

5.4 Materials and methods

Three sets of images shown in Figure 5-1 were used to illustrate how the UWT-MIA method performs textural and spectral segmentation under different scenarios.

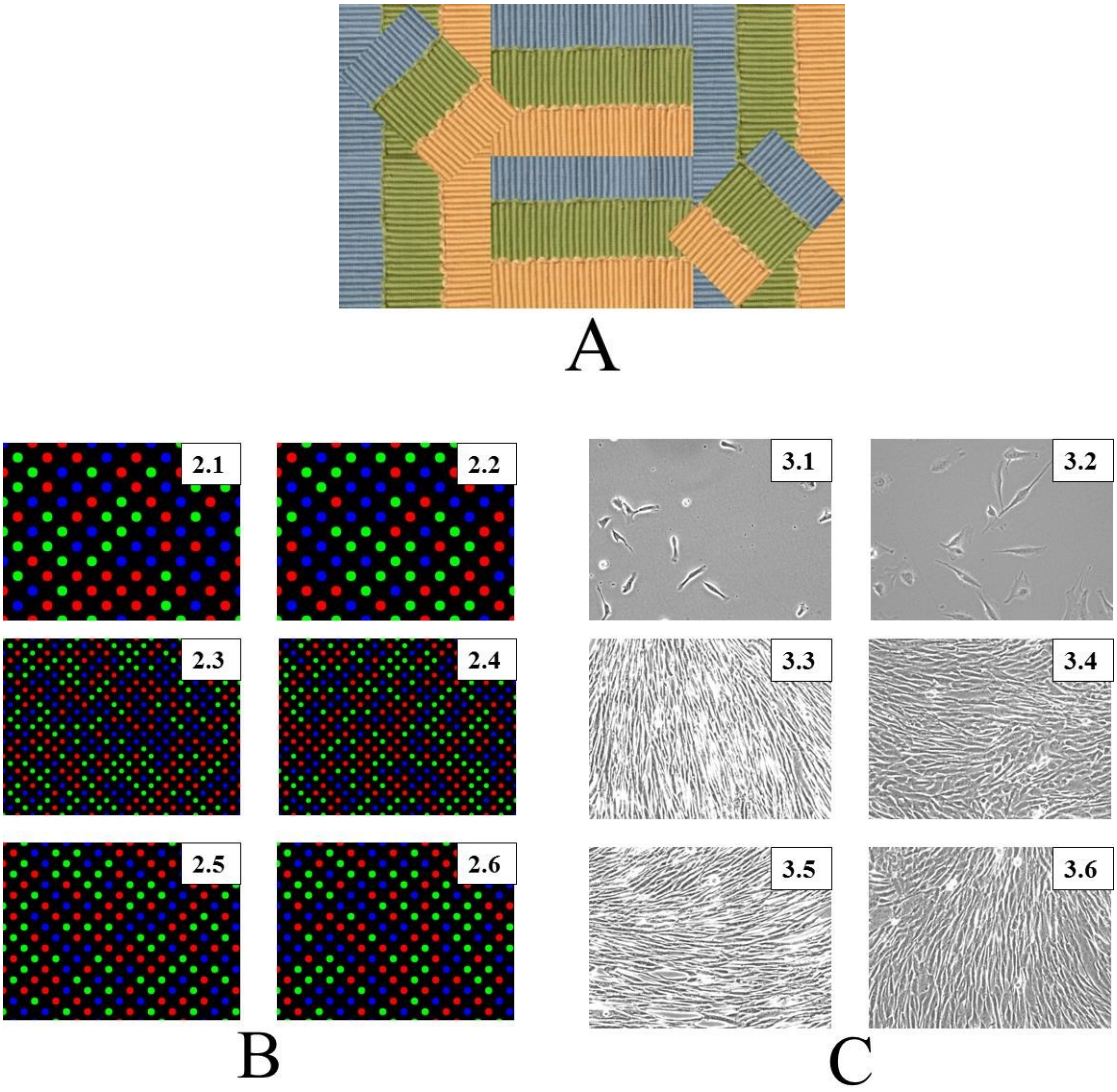


Figure 5-1 : Images used to illustrate the UWT-MIA approach. A) Set 1: one synthetic textured RGB image; B) Set 2: six synthetic RGB images with colored circular objects; C) Set 3: six phase contrast microscopy images of myoblasts.

In Set 1, a synthetic textured RGB image was generated from a source image freely available on the web (Kingdom and Olmos). Different textural regions were created by rotating the source image in several directions (mainly differing by their orientation). These regions also have different colors (spectral content).

Set 2 consists of six synthetic RGB images within which circular objects of 3 different sizes (small, medium, and large) and 3 colors (pure red, green, and blue) were randomly distributed in the image. Three series of two images were generated with objects of different sizes.

In Set 3, six typical greyscale images of myoblasts grown in a serum-supplemented (SSM) and a serum-free (SFM) medium were selected to perform a UWT-MIA analysis (at low and high degree of plate coverage by cells, or confluence). Myoblasts from a primary cell line (kindly provided by Dr. J.P. Tremblay, CRCHUL, Quebec City, QC, Canada) were cultivated in SSM (MCDB120 + 15% FBS) or in SFM (Parent 2009) in 6-well plates. Both media used a basal mixture, MCDB120, which is an optimized nutrient medium for human muscle cells (Ham and Blau 1988). Target inoculation density was in the range of 5,000-20,000 cells/mL, using a working volume of 2 mL/well. Growth medium was changed every 2-3 days. Phase contrast microscopy (PCM) images were captured with an IX-81 microscope (Olympus™) equipped with an automated stage and a Qimaging® monochrome camera at 20X. An incubator was mounted on the microscope stage to maintain the culture plate at 5% CO_2 and 37.5 ºC. The experimental set-up was controlled using Metamorph® (Molecular DevicesTM), and the focus was adjusted for each image using the autofocus utility. Images 3.1, 3.3 and 3.5 correspond to myoblasts grown in SFM, and images 3.2, 3.4 and 3.6 to myoblasts grown in SSM.

Each dataset was created (or selected) to illustrate the features of the UWT-MIA approach for different situations. Textural patterns in Set 1 are identical (same signal frequency content), but their orientation and their colors are different. In Set 2, both the size and the color of the circular objects vary. Finally, the third set includes variations in both the frequency content and orientation (i.e. cell size, shape and orientation), but there is only one single channel (grey level images).

5.5 The Undecimated Wavelet Transform – Multivariate Image Analysis (UWT-MIA)

The UWT-MIA method involves several steps to be performed on each raw image (Figure 5-2):

1. Calculate the Wavelet planes (WP) for each scale, orientation and wavelength;
2. Stack the Wavelet planes as a different channel of a multivariate image called the Wavelet Planes Cube (WPC);
3. Apply MIA on the WPC;
4. Obtain the score density histogram;
5. Segment the features of interest in the score space via masking.

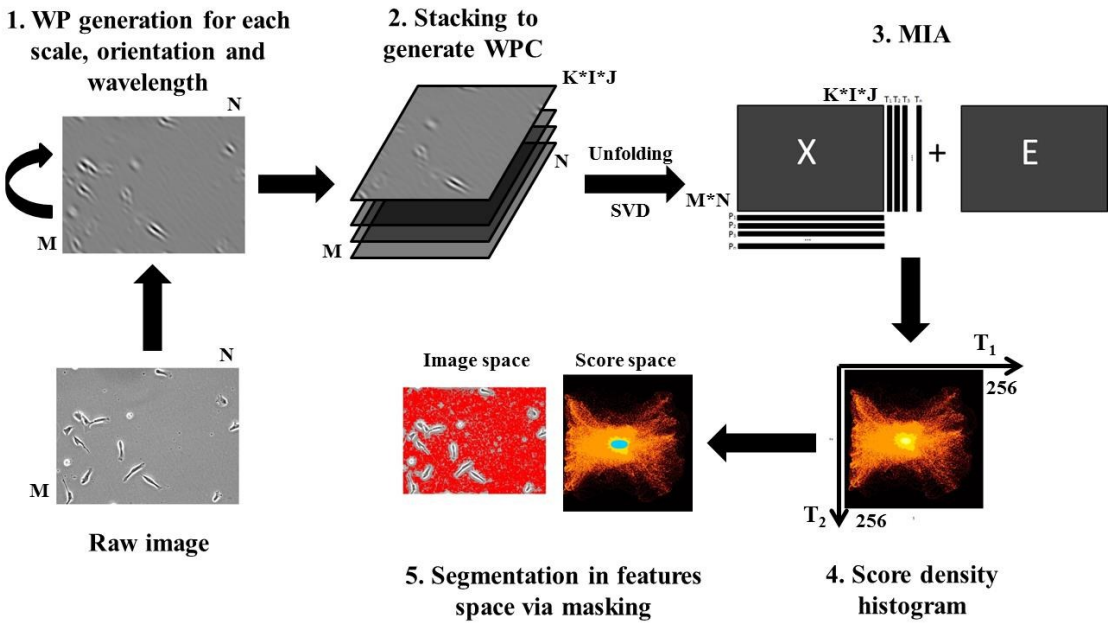


Figure 5-2 : Steps associated with the UWT-MIA method

This method can be extended to be applied on sets of images (e.g. time series) by using a kernel algorithm (Geladi and Grahn 1996). It involves computing the sum of the variance-covariance matrix of each unfolded WPC (one per image) followed by singular-value decomposition (SVD) to yield a common set of MIA loadings for all the images of the set.

5.5.1 Gabor wavelet planes generation

The Undecimated Wavelet Transform (UWT) analysis can be performed using two types of wavelets: the stationary wavelet transform (SWT) or the continuous wavelet transform

(CWT). The SWT computes wavelet planes in the image space with the *à trous* algorithm, which requires an upsampling of the wavelet filter by a factor of 2 at each scale (Starck, Elad et al. 2004). Alternatively, the CWT computes the wavelet planes in the Fourier domain (Jacques, Coron et al. 2007), and has a higher resolution (from a scale and orientation perspective) than the SWT. It was shown by Liu and MacGregor (2008) that the CWT results in a better resolution for estimating objects sizes since it has a constant resolution over all scales, as opposed to the DWT and SWT which produce wavelet planes that have fine resolution at low scales and coarse resolution at high scales due to dyadic sampling. To optimize both scale and orientation resolution, the Gabor filter (a CWT) was selected which shape also matches fairly well the signals of the images shown in Figure 5-1.

The two-dimensional Gabor filter is a frequency and orientation selective filter which consists of a sinusoidal plane wave modulated by a Gaussian envelope (Tuceryan and Jain 1999). Due to the robustness of their features against local distortions caused by variations of illumination, Gabor filters have been successfully applied for unsupervised texture segmentation in multi-channel filtering (Jain and Farrokhnia 1991), in face recognition (Shen and Bai 2006) and in mineral flotation process monitoring (Liu, Gui et al. 2013). Gabor filters/wavelets seem to be the optimal basis to extract local features for pattern recognition, since the shape of the Gabor wavelet is similar to the receptive fields of simple cells in the primary visual cortex (Shen and Bai 2006). Gabor wavelets are optimal for measuring local spatial frequencies, and they yield distortion tolerant features for other pattern recognition tasks such as handwritten numeral and fingerprint recognition (Shen and Bai 2006). The Gabor wavelets also exhibit desirable characteristics of spatial frequency, spatial locality and orientation selectivity (Shen and Bai 2006). Although the Gabor function is an admissible wavelet, it does not result in an orthogonal decomposition, meaning that the Gabor wavelet transform is redundant, generating correlated wavelet planes (Jain and Farrokhnia 1991).

Intuitively, a wavelet transform can be conceptualized as a band-pass filtering operation on the input image (Jain and Farrokhnia 1991). The Gabor filter is a band-pass oriented filter situated in the Fourier domain and delocalized from the zero frequency, thus generating a sinusoidal wavelet. The filter associated with the Gabor wavelet (ψ_{Gabor}) can be computed directly in the Fourier domain (without going through the spatial domain) for different scales s and angles θ using the Yawtb approach (Jacques, Coron et al. 2007). The Gabor

wavelet filter is described by equations 5-1 to 5-3, where *u* and *v* are the horizontal and vertical frequencies in the image expressed in rads/pixel:

$$\psi_{\mathrm{Gabor}}(u, v, \sigma, \varepsilon, k_0, s, \theta) = \exp(-\sigma^2((u_r - k_0)^2 + (\varepsilon v_r)^2)/2) \quad (5\text{-}1)$$

$$u_r = s * \left(u * \cos(\theta) - v * \sin(\theta)\right) \quad (5\text{-}2)$$

$$v_r = s * \left(u * \sin(\theta) + v * \cos(\theta)\right) \quad (5\text{-}3)$$

Parameters *s* and *θ* correspond to the scale and orientation of the wavelet, respectively. The width of the Gaussian envelope in the Fourier domain *σ* controls the sharpness of the wavelet. The lower the value of *σ* the sharper the wavelet is (useful for edge detection, for example). The length of the wavelet in the spatial domain is adjusted through parameter ε. It modulates the trade-off between the spatial and frequency resolutions. Increasing ε leads to a longer wavelet which improves the frequency resolution at the expense of a lower spatial resolution (i.e. reduced spatial localization ability) and vice-versa. The offset parameter k_0 also changes the wavelet shape and the frequency content by introducing an offset from the zero frequency. Increasing k_0 leads to a more complex wavelet shape (i.e. richer frequency content). These parameters need to be selected in such a way to best match the wavelet shape to that of the signal. Unfortunately, there are no explicit and well defined guidelines for doing this. These parameters can be roughly tuned by trial-and-error or grid search until satisfactory results are obtained. Several examples of Gabor wavelet filters in the spatial domain are provided by Shen et al. (2006). Note that for angles *θ* = 0 and 4π/4 the Gabor wavelet is oriented vertically (capturing vertical patterns) and for angles *θ* = 2π/4 and 6π/4 the Gabor wavelet is oriented horizontally (capturing horizontal patterns), as illustrated in Figure 5-3.

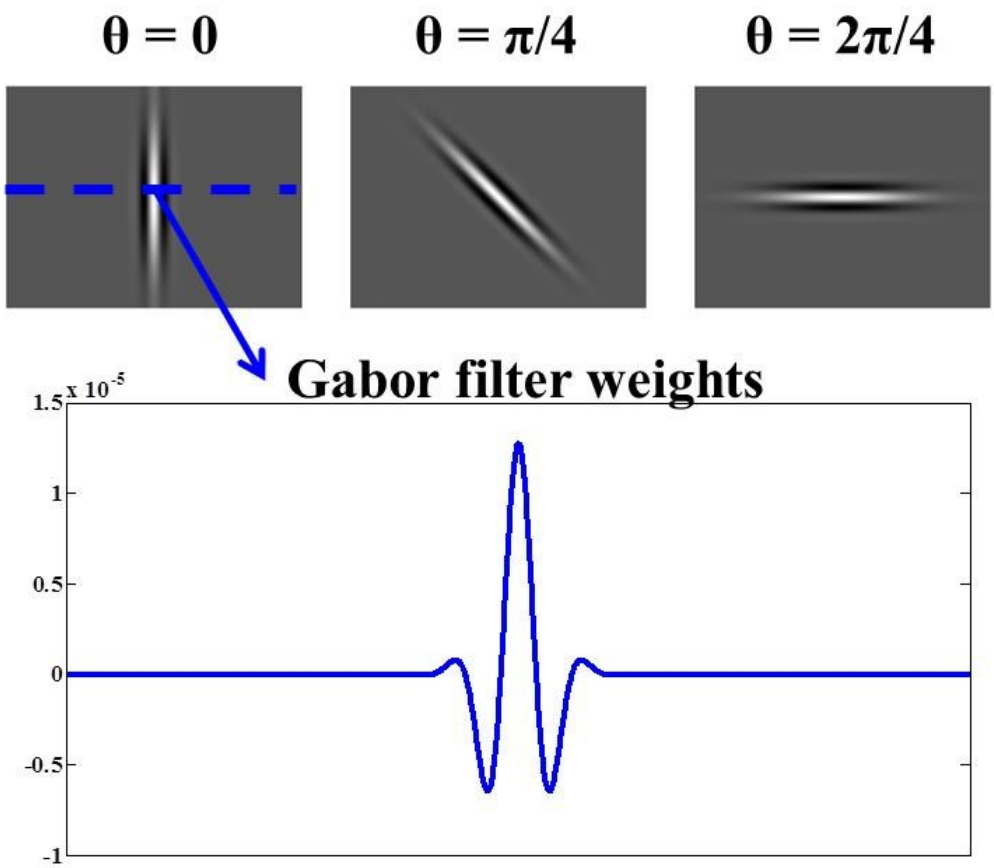


Figure 5-3 : The 1D and 2D shape of the Gabor wavelet for a given set of parameters and orientation angles (*θ* = 0, π/4 and 2π/4). White pixels are associated with positive filter weights and black pixels to negative filter weights.

The convolution between the Gabor filter and the raw image is performed directly in the Fourier domain in order to be computationally more efficient than in the spatial domain (Shen and Bai 2006). The result of the convolution is translated back into the spatial domain (Gonzalez and Woods 2008). The pixel-by-pixel convolution does not involve downsampling or upsampling. Thus, the Gabor wavelet is a continuous wavelet transform. Also, since all the wavelet planes have the same size as that of the original image, it is considered an undecimated wavelet transform (UWT). The convolution operation to generate each wavelet plane (WP) from a raw grey scale image I(x,y) is given by equation 5-4. Point-by-point multiplication of the complex conjugate of the wavelet filter in the Fourier domain (ψ_{Gabor}) and the Fourier transform of the raw image (FFT(I(x,y))) is

performed first, followed by calculating the inverse Fourier transform of the convolution (keeping only the real part) (Jacques, Coron et al. 2007).

$$\mathrm{WP}_{\mathrm{Gabor}}(x, y, \sigma, \varepsilon, k_0, s, \theta) = \mathrm{real}\left(\mathrm{IFFT}\left[\mathrm{FFT}\left(I(x, y)\right).* \mathrm{conj}(\psi_{\mathrm{Gabor}}(u, v, \sigma, \varepsilon, k_0, s, \theta))\right]\right) \quad (5\text{-}4)$$

If the image to be analyzed is multivariate (e.g. RGB or multi-spectral images), the convolution is performed independently on the grey scale image corresponding to each wavelength as in the MR-MIA I approach (Liu and MacGregor 2007). However all the wavelet planes from all the wavelengths can be stacked together into a single array prior to performing MIA, as illustrated in the next section.

5.5.2 UWT-MIA

Since the wavelet planes have the same size, it is possible to stack the sub-images obtained for each of the *I* scales ($s = s_1, s_2 \ldots s_I$), *J* angles ($\theta = \theta_1, \theta_2 \ldots \theta_J$) and *K* wavelengths ($\lambda = \lambda_1, \lambda_2 \ldots \lambda_K$) as a separate channel in a new multivariate image, as illustrated in Figure 5-4. This multivariate image is referred to as the Wavelet Planes Cube (WPC) in the rest of the paper.

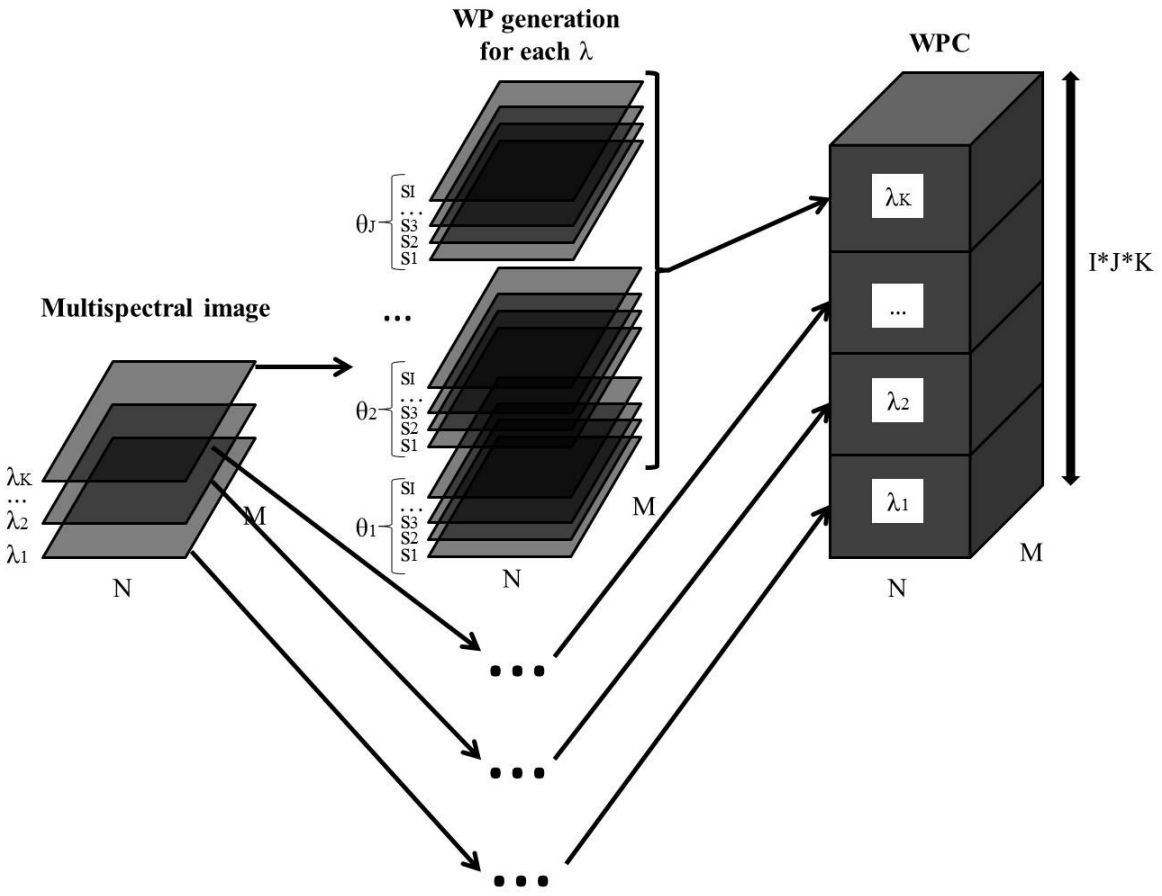


Figure 5-4 : Generation and stacking of the wavelet planes (WP) obtained for each wavelength (λ), angle (θ), and scale (s) into a cube (WPC)

Multivariate Image Analysis (MIA) can be applied for spatial-spectral analysis of the WPC in the very same way as for a typical multivariate image (Geladi and Grahn 1996; Lied, Geladi et al. 2000). That is, the 3-way array is unfolded into a matrix **X** of dimensions (N*M)×(I*J*K) where the columns correspond to wavelet plane coefficients at each scale, angle and wavelength for each pixel (rows), as shown in Figure 5-2. Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) is then applied to the unfolded matrix **X**. The score scatter plots based on a selected number of principal components (*A*) are shown as density histograms on which a masking/overlay procedure is applied for exploring the image information (Geladi and Grahn 1996). When MIA needs to be applied to a set of images, a common latent variable space can be obtained using the kernel algorithm published in Geladi and Grahn (1996), where SVD is applied to the sum of the

variance-covariance matrix of each image. Pixel groups belonging to similar textural and spectral profiles will cluster together in a region of the $\mathbf{T}_1/\mathbf{T}_2$ score space (or other pairs of components), and can be segmented by using an appropriate mask. The masks can be determined manually by trial-and-error (traditional approach), or by an automatic supervised classification method such as SVM (Liu, Bharati et al. 2005). When the objective is to use the image information for predicting process or response variables of interest **Y** (i.e. multivariate image regression problems), this second set of data can be used to identify more specific masks using only the most relevant information for predicting **Y**. The number of masks (i.e. pixel groups) to use can also be determined by trial-and-error or by an automated procedure combing unsupervised and supervised classifiers to extract the number of classes in an iterative manner (Noordam, van den Broek et al. 2005). All of these approaches can be used to extract information from the features space (i.e. score density histograms) obtained by UWT-MIA. In this work a trial-and-error method is used to manually draw several masks to explore the score space and see what type of information is related to the main pixel clusters. However any method developed for MIA masking would be applicable.

All image processing operations were performed in Matlab® (Image Processing Toolbox®) using custom scripts. To compute the continuous Gabor wavelet planes, Matlab® scripts from the Yawtb group (Jacques, Coron et al. 2007) were adapted for this application. Note that the scores values were scaled between 0-255 (min-max scaling), rounded and binned to construct the score density histograms (256 x 256 bins). The origin of the score density histograms (bin associated with the minimum score values) was set to be at the top left corner of the plots. This needs to be taken into consideration when interpreting the clustering patterns within the score spaces.

5.6 Results and Discussion

The UWT-MIA was performed on each set of images using the Gabor wavelet parameters listed in Table 5-1. The parameters were set by trial-and-error to provide meaningful textural information in order to segment important patterns within images. Note that in all three case studies, it was decided to apply no centering or scaling to the unfolded matrix **X** prior to performing PCA because auto-scaling led to a very similar interpretation. However, it is always recommended to verify the impact of scaling as part of the image analysis process. The following sections summarize and discuss the results obtained for each dataset.

Table 5-1 : Gabor wavelet parameters used for each image set

Gabor wavelet parameters		Set 1	Set 2	Set 3
Number of wavelengths (K)		3	3	1
s	Min	1	1	1
	Max	40	40	50
	Interval	1	1	1
	Number of scales (I)	40	40	50
θ	Min	0	0	0
	Max	$7\pi/4$	$7\pi/4$	$7\pi/4$
	Interval	$\pi/4$	$\pi/4$	$\pi/4$
	Number of angles (J)	8	8	8
σ	-	0.5	0.5	0.5
ε	-	5	1	1
k_0	-	5	5	5
Total number of WPs		960	960	400

5.6.1 Textural and spectral segmentation of a RGB image (Set 1)

The results obtained after applying UWT-MIA to the synthetic textured RGB image ($1152\times1920\times3$) in Set 1 are shown in Figure 5-5. The wavelet decomposition was applied to each wavelength of the RGB image (K = 3, λ =Red, Green, Blue) using a total of 40 scales (s = 1, 2, …,40) and 8 angles (θ = 0, $\pi/4$, …,$7\pi/4$), which yields a 3-way array (WPC) of dimensions $1152\times1920\times960$. Although additional PCA components capture useful information, the first two were found sufficient for the purpose of illustrating how UWT-MIA extracts the color and orientation of the textural patterns (the frequency content of these patterns is similar across the image). The percentage of variance explained by the first two components was 84.6% and 4.7%, respectively.

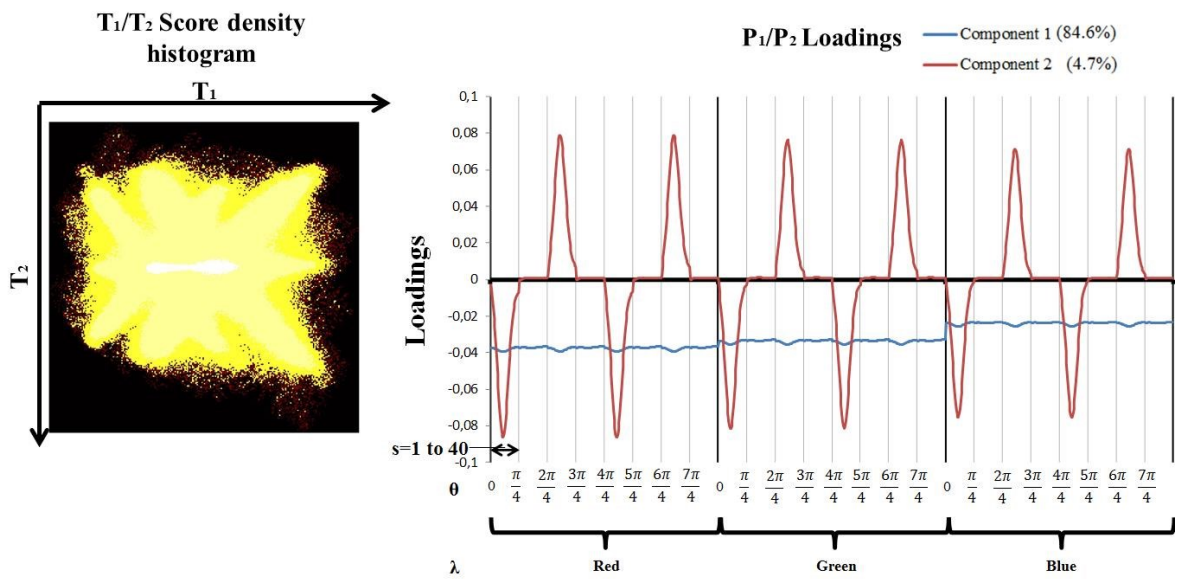


Figure 5-5 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 1 image

The loading plot shown in Figure 5-5 reveals that the first component captures variations in light intensities associated with the three RGB color channels since all loadings have the same sign. Considering that the loading values are all negative, it is expected that the yellow, green, and blue pixels should be distributed from left to right along the $\mathbf{T}_1$ axis since yellow is the brighter color, blue is the darker color and green has intermediate intensity values. Thus, the first score image is just a grey level version of the RGB image, which is a typical result obtained with MIA (Duchesne, Liu et al. 2012). On the other hand, the second component is clearly related to the orientation of the textural patterns in the image. The loading values are positive for horizontal patterns (θ = 2π/4 and 6π/4), negative for vertical patterns (θ = 0 and 4π/4), and near zero for diagonal patterns. Note that the relationship between the angles and the orientation of the patterns is counter-intuitive. However, this is in agreement with the spatial orientation of the Gabor wavelet as discussed in Section 5.5.1 and illustrated in Figure 5-3. It is also important to mention that additional PCA components capture useful information, but were not shown in the article for sake of conciseness. For instance, the fourth component captures the contrast between the red and blue channels. Combining $\mathbf{T}_1$ and $\mathbf{T}_4$ allows segmenting the yellow, green and blue textural patterns independently.

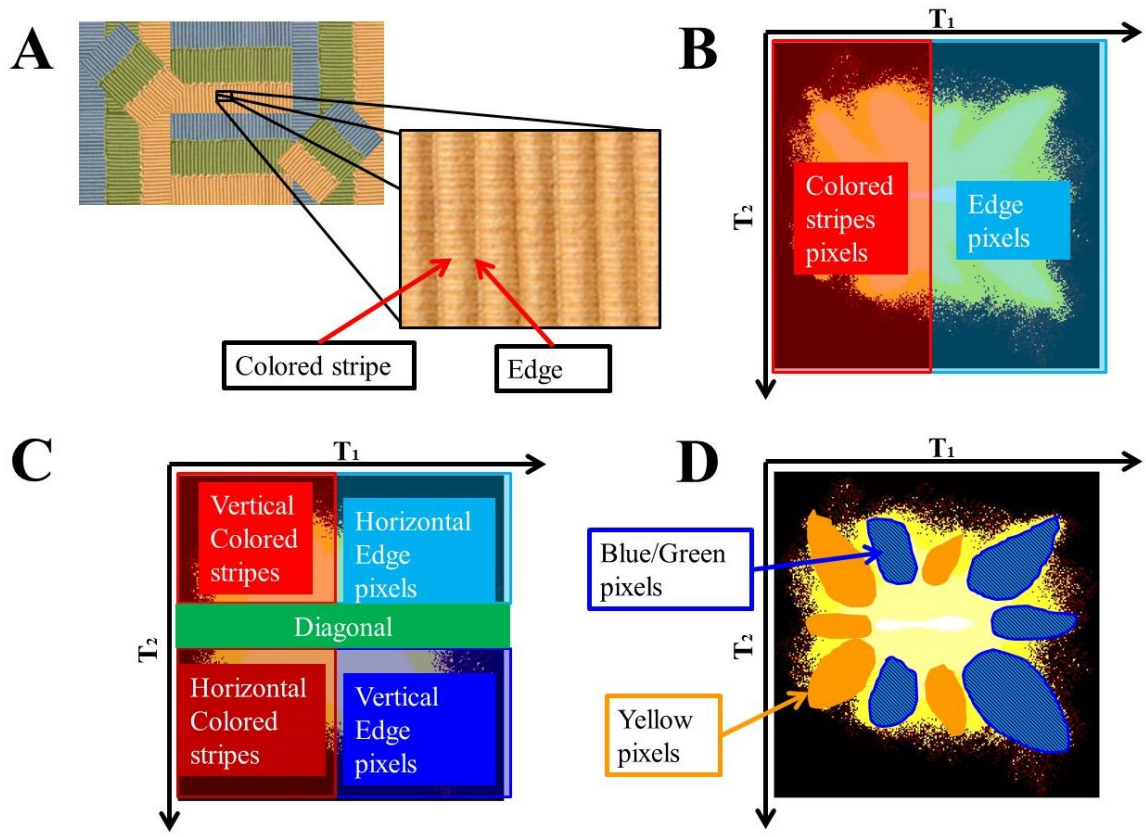


Figure 5-6 : Investigation of the pixel clusters in T_1/T_2 score density histogram. A) Raw image showing colored stripes and edges. Localization of pixels belonging to B) colored stripes and edge pixels and C) vertical, horizontal and diagonal patterns. D) Clustering of blue/green and yellow pixels.

To investigate the information contained in the image, different regions of interest are explored in the $\mathbf{T}_1/\mathbf{T}_2$ score space, as shown in Figure 5-6. Pixels distribute along the $\mathbf{T}_1$ axis according to the color of the textural region they belong to, but also depending upon whether they lie on colored stripes or edges (darker lines between colored stripes, see Figure 5-6A). Darker edge pixels have lower WP coefficients and, because of the negative $\mathbf{T}_1$ loadings, they fall in the right hand side of the score density histogram. The brighter colored stripes pixels have higher WP coefficients and fall in the opposite side of the score plot. This is consistent with the information captured by $\mathbf{T}_1$, acknowledging that lighter blue stripes pixels might overlap slightly with darker yellow edge pixels in terms of grey level intensities. However, these ambiguities can be resolved by using $\mathbf{T}_1$ and $\mathbf{T}_4$ simultaneously in the segmentation process (not shown). Along the second component (i.e. $\mathbf{T}_2$ direction), the pixels cluster according to the orientation of the textural region they are associated

with. First, it is important to mention that, in general, edge pixels yield negative WP coefficients while these are positive for colored stripes pixels. This is due to the fact that the Gabor filter has positive weights at its center (white pixels) and negative weights at its borders (black pixels) as shown in Figure 5-3. When the Gabor filter passes over a colored stripe, its positive weights are convoluted with strong grey level values therefore resulting in positive wavelet coefficients. The opposite situation occurs when the filter passes over edge pixels: the negative weights which are now over the adjacent colored stripes (strong grey level values) yield negative wavelet coefficients. Hence, the colored stripe signals are positively correlated with the Gabor wavelet whereas the edge signals are negatively correlated. Examples of WPs for Set 1 are provided in Figure 5-7. Considering this and the signs of the $\mathbf{p}_2$ loadings (Figure 5-5), pixels associated with horizontal colored stripes fall in the bottom-left corner of the score density histogram (high $\mathbf{T}_2$ scores values) and vertical colored stripes fall in the upper-left corner (low $\mathbf{T}_2$ scores values). For edge pixels having negative WP coefficients, the clustering pattern is exactly the opposite (Figure 5-6C). Since pixels belonging to diagonal patterns are associated with a mixture of horizontal and vertical signals, the diagonal information can be found in the middle range of $\mathbf{T}_2$. Finally, since yellow pixels have higher grey level values associated to the red wavelength (in the RGB color space), it is possible to find groups of yellow pixels always to the left of their green/blue counterparts (Figure 5-6D) along the $\mathbf{T}_1$ axis.

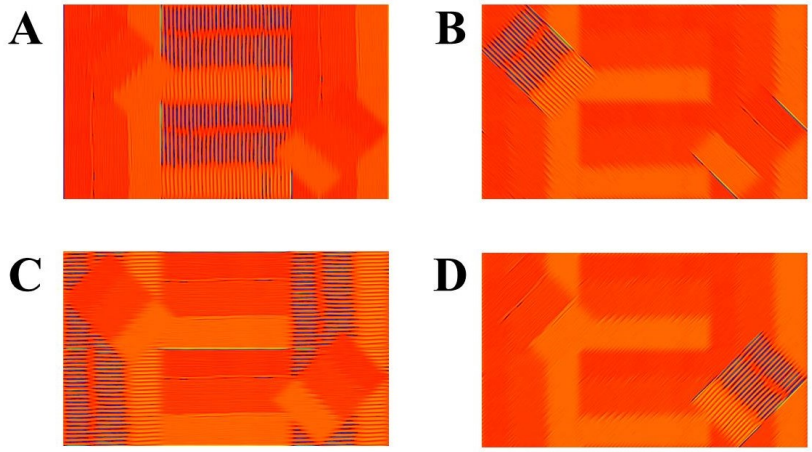


Figure 5-7 : Pseudo-color images of WPs for Set 1 for λ = Red, s = 10 and for A) θ = 0, B) θ = π/4, C) θ = 2π/4, D) θ = 3π/4. In the color map, red/yellow colors correspond to positive wavelet coefficients and blue/green colors to negative wavelet coefficients.

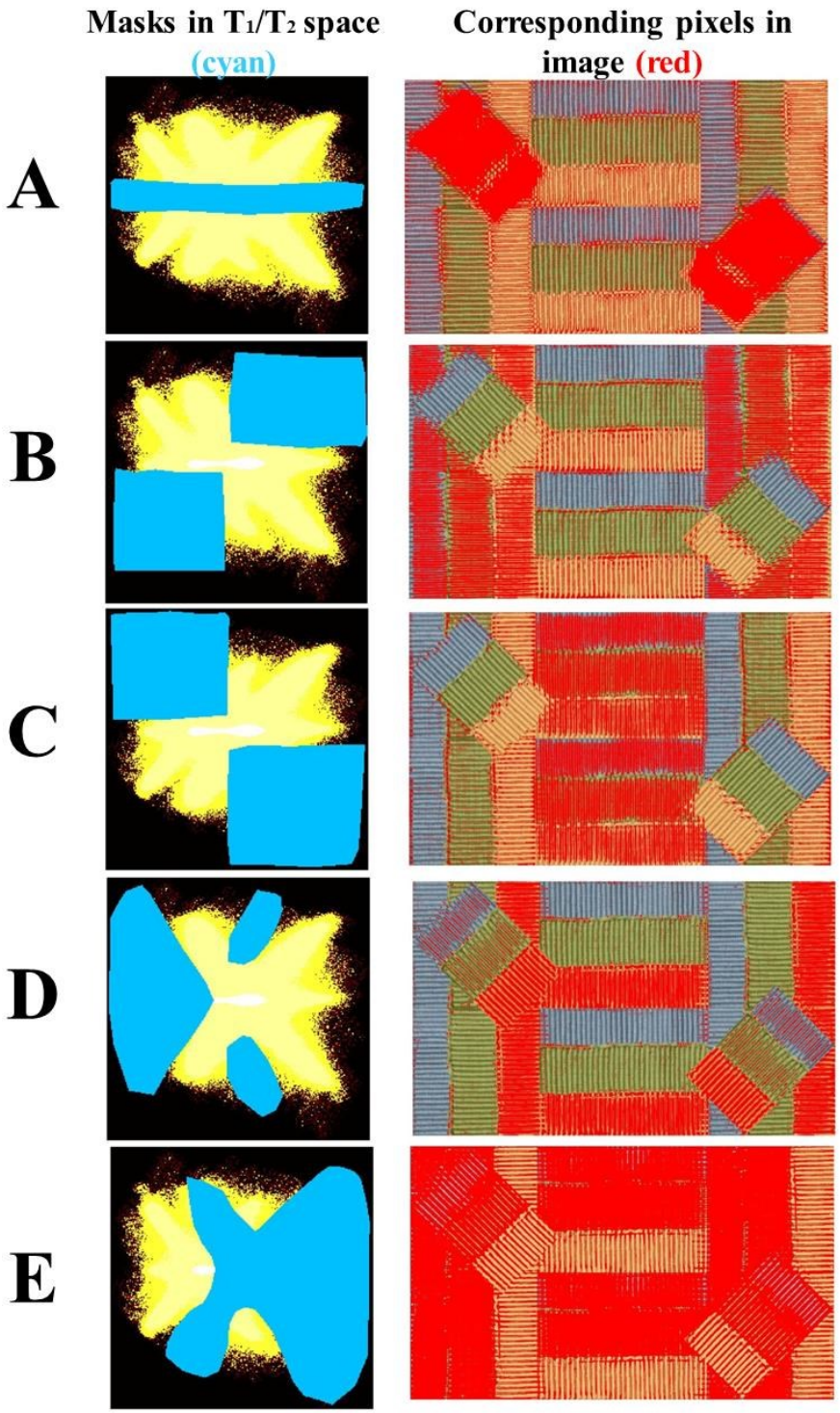


Figure 5-8 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in red) for Set 1. Masks are used to select A) diagonal, B) horizontal, C) vertical, D) yellow, and E) blue/green pixel regions.

The UWT-MIA masking procedure is illustrated in Figure 5-8. Various masks have been drawn to segment groups of pixels belonging to diagonal (A), vertical (B) and horizontal (C) texture regions independently of their color (spectral content). The last two masks were selected in such a way to segment groups of pixels associated with the yellow (D) and green/blue (E) regions, independently of the orientation of the textural patterns. This shows the ability of UWT-MIA to perform spectral-texture segmentation of multispectral images using a single MIA model.

5.6.2 Textural and spectral segmentation in multiple RGB images (Set 2)

Set 2 consists of 6 RGB images of dimensions (1200×1600×3), two images for each object size (randomly spatially distributed in the image). Again, two types of variations are present in the images: the objects have different colors and sizes. The latter changes the frequency content within the images, which should be captured by the wavelet scales. Since the objects are perfectly circular, the UWT-MIA approach should not detect any preferential orientation. That is the loading values for the various angles at a given scale and wavelength should be similar. The UWT-MIA algorithm was applied to each image separately using 40 scales and 8 angles as in the previous case study (Set 1). After unfolding the six (1200×1600×980) WPCs (one per image), the kernel algorithm (Geladi and Grahn 1996) was used to build a single MIA model (i.e. a common latent space for all images). The first two components explain about the same amount of variance in the image (19.6% and 18.8%, respectively). The loading plots and the $\mathbf{T}_1/\mathbf{T}_2$ global score density histogram are shown in Figure 5-9.

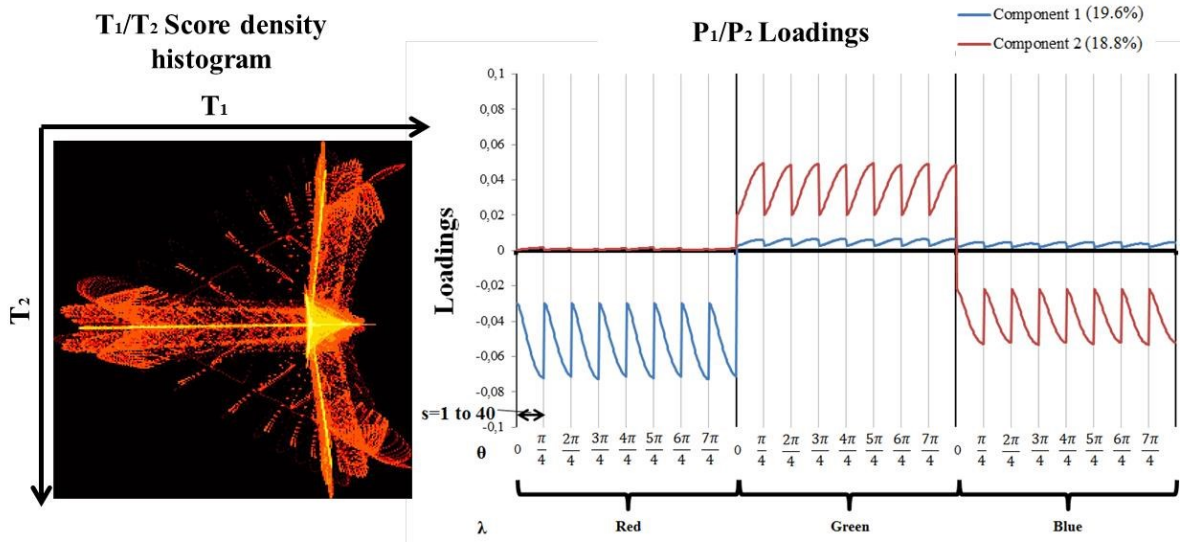


Figure 5-9 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 2 images

The signs of the loading values within each of the first two components clearly indicate that they capture the color of the objects. The first component explains the red color (contrast between the red and the other colors) and the second models the difference between the blue and the green colors. For both components, the periodic trend in the loadings indicates that from a textural perspective there is no preferential orientation because the loading values for a given color and scale are the same for all angles. However, this periodic trend also shows that loading values are smaller at lower scales (associated with higher frequencies or smaller objects) and larger at higher scales (lower frequencies or larger objects). It is therefore expected that the variability within each score vector is associated with the size of the objects.

The $\mathbf{T}_1/\mathbf{T}_2$ score density histogram also shown in Figure 5-9 has four main regions of interest: red pixels fall in the region of low $\mathbf{T}_1$ values, blue pixels fall in the upper right (high $\mathbf{T}_1$ and low $\mathbf{T}_2$ values) region, and green pixels cluster the lower right region (high $\mathbf{T}_1$ and $\mathbf{T}_2$). Finally, background pixels (black color) fall in the center right region of the score plot. The variability within each region is not related with color variations (pure red, blue and green were used) but rather with the size of the objects (associated with the frequency content). The further we are from the center right of the plot (background pixels region), the smaller the objects captured are. This is mainly due to the fact that pixels belonging to smaller objects generate consistently higher positive WP coefficients at all scales compared to larger objects as a result of the convolution with the Gabor wavelet. Larger

objects are captured completely and only by wavelets at higher scales, as opposed to smaller objects which trigger a filter response at both low and high scales (as the Gabor wavelet will detect these small objects as an edge detector). Thus there is a cumulative effect discriminating smaller from larger objects.

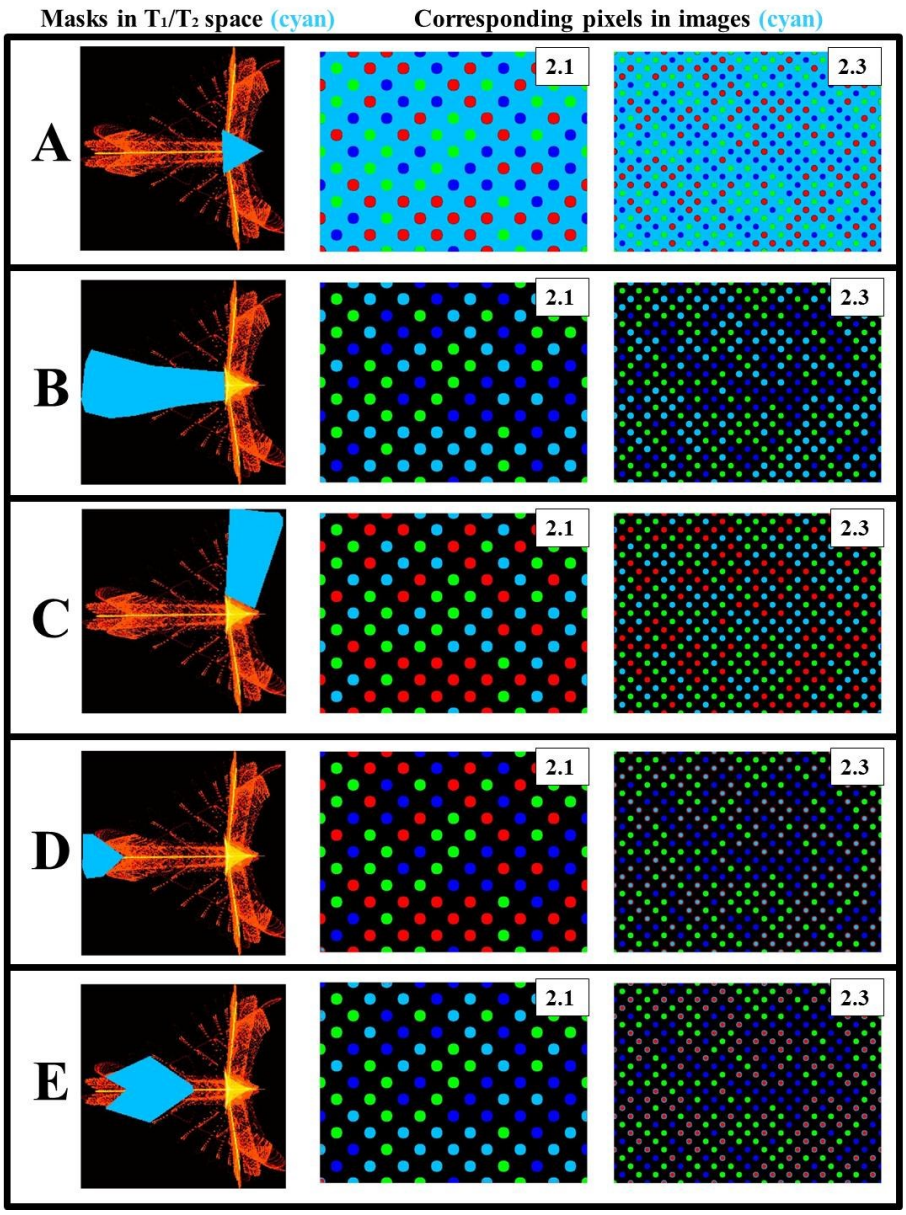


Figure 5-10 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in cyan) for Set 2. Selected pixels correspond to A) background region, B) red objects, C) blue objects, D) small red objects, and E) large red objects.

The masking procedure for Set 2 images is shown in Figure 5-10. Since the loadings of the first principal component are negative for the red color, red pixels cluster in the left hand side of the score density histogram (low $\mathbf{T}_1$ scores), with smaller objects being at the extreme left since they have higher positive WP coefficients at all scales. The same explanation holds for the second component. Green pixels fall in the bottom-right corner since they are associated with positive $\mathbf{p}_2$ loadings (high $\mathbf{T}_2$ scores), whereas blue pixels are located in the upper-right section because of the negative $\mathbf{p}_2$ loadings (low $\mathbf{T}_2$ scores). Since they have higher positive WP coefficients compared to bigger objects, small blue objects fall in the top right and small green objects in the bottom right. The masks shown in Figure 5-10 select pixels associated with the black background (A) and pixels belonging to red (B) and blue (C) objects, respectively. Figures (D) and (E) illustrate that the red pixels projecting further away from the center of the score plot belong to progressively smaller objects.

5.6.3 Texture segmentation in multiple cellular images (Set 3)

The set of 6 grey level images of myoblasts (Set 3) was processed similarly as for Set 2, except that 50 scales were used instead of 40 (same number of angles). This yielded ($1200\times1600\times400$) dimensional 3-way arrays (I=50, J=8, K=1). Again, a similar approach was used to build a single MIA model on this set of images, which contained three sources of variation: changes in light intensity, size and orientation of the cells (objects). The $\mathbf{T}_1/\mathbf{T}_2$ score density histogram for all 6 images as well as the corresponding loadings plots are shown in Figure 5-11. Both components explain 78.1% and 6.5% of the variance, respectively.

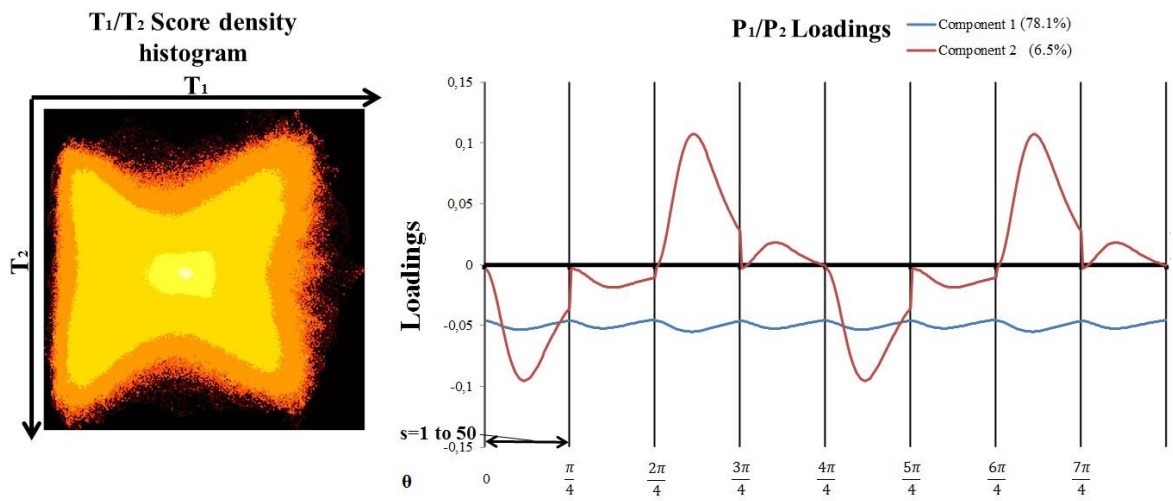


Figure 5-11 : T_1/T_2 Score density histogram and P_1/P_2 loadings for Set 3 images

The loadings of the first component capture variations in light intensities (grey levels) across the images (loading values are all negative) in addition to smaller variations that depend on scale. The latter are replicated for the different angles. The second component combines variations in both the scale and orientation of the textural patterns (objects) within the images. The loadings are positive for horizontal patterns (θ = 2π/4 and 6π/4) and negative for vertical patterns (θ = 0 and 4π/4). Again, diagonal patterns correspond to near zero loading values.

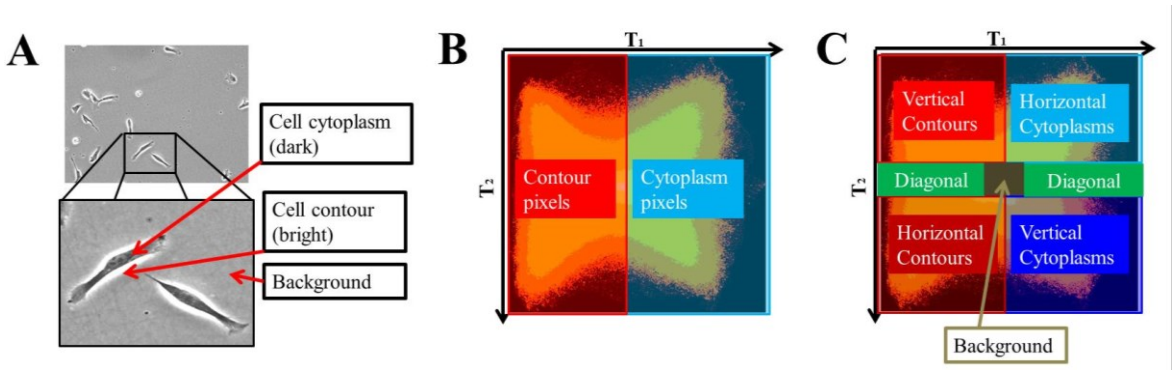


Figure 5-12 : Investigation of the pixel clusters in T_1/T_2 score density histogram. A) raw image showing the cells cytoplasm and contour pixels. Localization of pixels belonging to B) cell contour and cytoplasm, C) vertical, horizontal, diagonal and background pixels.

As can be seen in Figure 5-12A, the cell contour appears brighter than the cytoplasm due to the "halo" effect in phase contrast microscopy (Otaki 2000), and this has an impact on

the grey level values of the pixels and also on the WP coefficients. As a result of the convolution with a Gabor wavelet, pixels located within the bright contours of the cells have positive WP coefficients as opposed to pixels associated with the cells cytoplasm which have negative WP coefficients. As discussed for Set 1, the sign of the WP coefficients indicate whether the correlation of the image signals with the wavelet is positive or negative. Since the first component has negative loadings, pixels associated to the cells cytoplasm project in the right-hand side of the score density histogram (high $\mathbf{T}_1$ scores) and pixels associated with the contours fall in the left-hand side (Figure 5-12B). Since the loadings capturing vertical patterns in the second component are negative, vertical contour pixels cluster in the top-left corner of the score density histogram (low $\mathbf{T}_2$ scores) whereas pixels belonging to vertically oriented cytoplasm are found in the bottom-right corner (high $\mathbf{T}_2$ scores, Figure 5-12C). For horizontal patterns the relationship is opposite because they are associated with positive loadings: horizontal contour pixels cluster in the bottom-left corner (high $\mathbf{T}_2$ scores) and the horizontally oriented cytoplasm pixels fall in the top-right corner (low $\mathbf{T}_2$ scores). Since background regions typically contain low frequency information, corresponding pixels fall in the middle of the score density histogram. Finally, pixels belonging to diagonal patterns are associated with a combination of horizontal and vertical signals. Thus, diagonally oriented objects are found in the middle range of $\mathbf{T}_2$. To better understand the impact of cell density and orientation on the clustering pattern of the pixels in the score space, a set of three individual images were projected on the global PCA model and the results are shown in Figure 5-13. As cell confluence increases, a greater number of pixels move from the middle of the $\mathbf{T}_1/\mathbf{T}_2$ score density histogram (e.g. image 3.1 in Figure 5-13) and outward (e.g. images 3.3 and 3.5 in Figure 5-13). As confluence increases and myoblasts grow, they become more elongated and oriented, and pixels tend to cluster in the top-left and bottom-right corners of the $\mathbf{T}_1/\mathbf{T}_2$ score density histogram for vertical cells (e.g. image 3.3 in Figure 5-13) and in the bottom-left and top-right corners for horizontal cells (e.g. image 3.5 in Figure 5-13). In this particular case, knowing the exact orientation of the myoblast is not so much important as detecting their global tendency to become preferentially oriented because this is an indication of the unset of cell fusion to form myotubes, and eventually muscle tissue (i.e. orientation is related with cell functionality). However, current studies on the anti-inflammatory response of certain types of cells under flow conditions involve measuring how they orient themselves with respect to flow direction. It is also easy to imagine that the

specific orientation of certain types of defects on the surface of a product might be useful information for quality control and defect diagnosis.

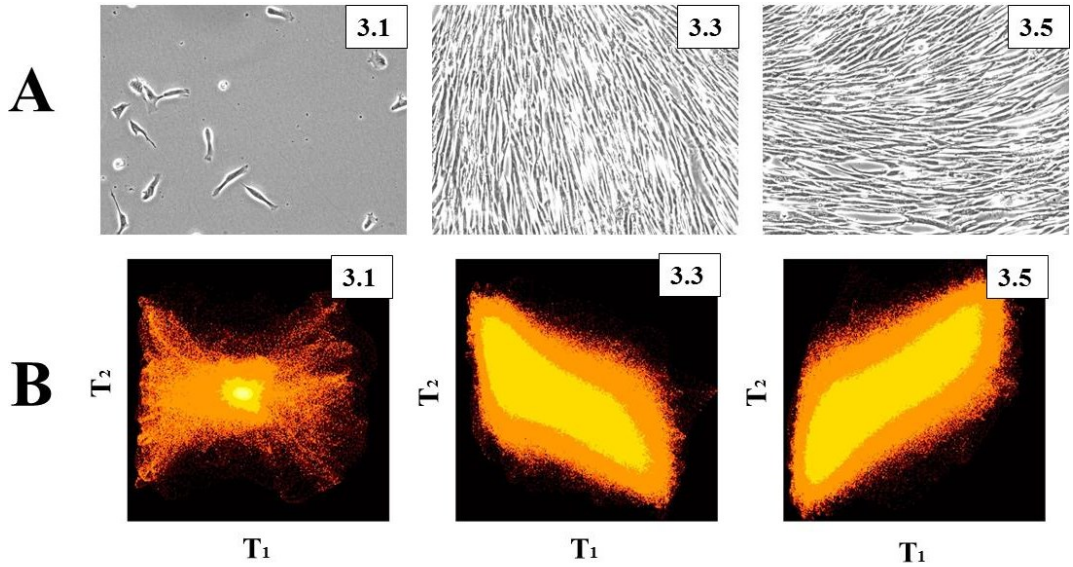


Figure 5-13 : T_1/T_2 individual score density histograms for three images of Set 3. A) Raw images, B) corresponding individual score density histograms. The effect of the degree of confluence (number of cells) and the cells orientation on the T_1/T_2 score density histogram are shown.

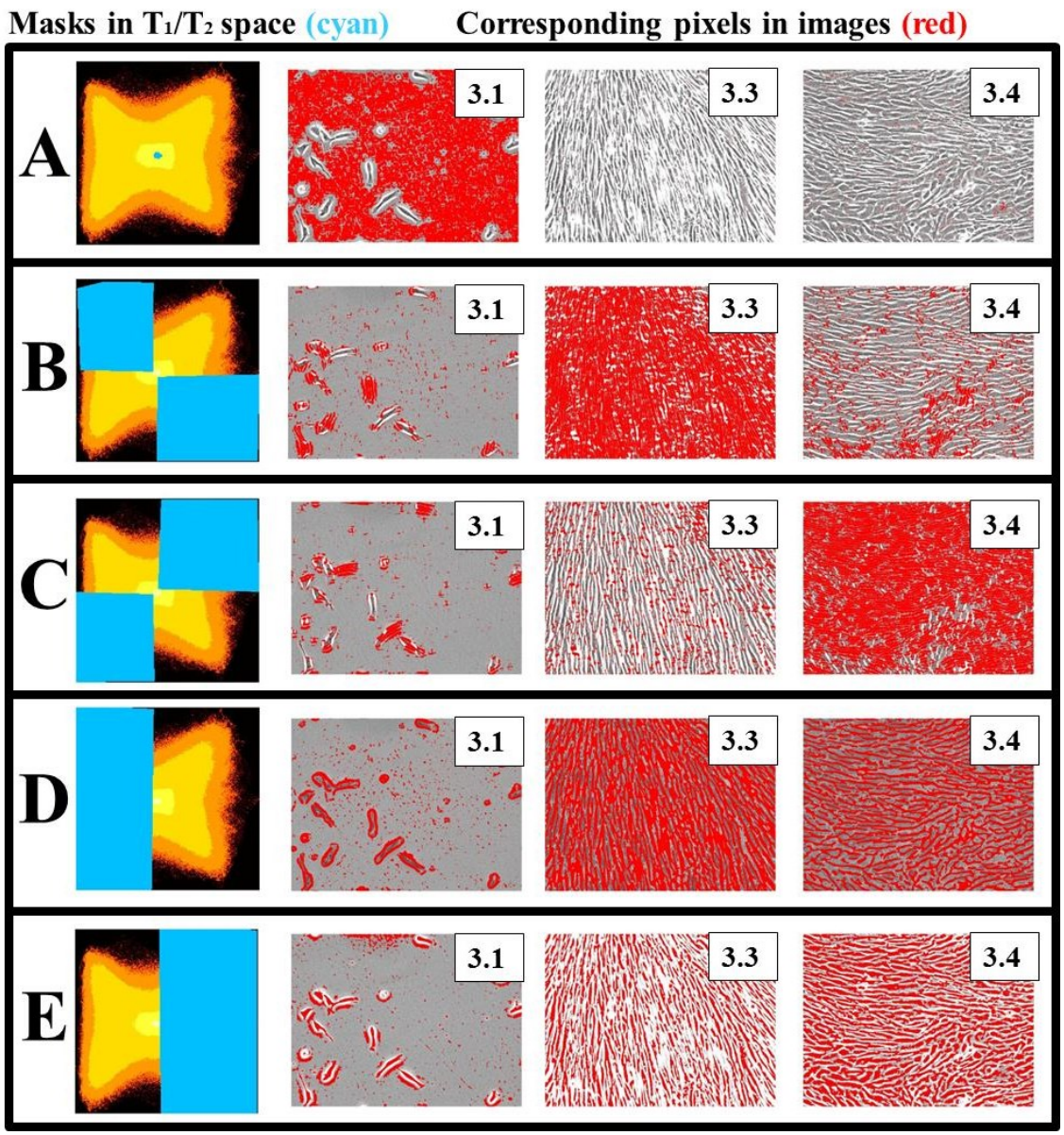


Figure 5-14 : Masking of groups of pixels in the T_1/T_2 score space (in cyan), and overlay of the selected pixels in the raw image (in red) for Set 3. Selected pixels correspond to A) background regions, B) vertical cells, C) horizontal cells, D) cellular contours, and E) cells cytoplasm.

The information contained in the cellular images is explored in Figure 5-14 using a masking/overlay procedure. The masks were selected in order to extract features related with level of confluence and cellular orientations. The mask drawn in the center of the score density histogram corresponds to background pixels (A), which have a medium grey level value and belong to low frequency regions within the image (compared to cellular

regions). The second and third rows of images show masks capturing pixels belonging to cells that are oriented vertically (B) or horizontally (C). The cells contours and cytoplasm masks are shown in row (D) and (E), respectively. Clearly, the light intensity-texture segmentation provided by the UWT-MIA approach could be used to track the level of confluence (number of cells) as well as the cell size and their orientation as a function of time during cell culture experiments. All of these features are physically meaningful for cell culture engineers.

In the field of phase contrast microscopy, the UWT-MIA approach could be used to monitor cellular morphology and orientation in high throughput screening and process control applications. Cells respond to environmental signals that regulate their behaviors (cell differentiation, division, migration or fusion), and scientists and engineers study changes in their morphology to understand the influence of those signals (Theriault, Walker et al. 2012). Since myoblasts at high confluence tend to become oriented just before undergoing fusion and differentiation into myotubes or myofibers (Freshney 2000), textural segmentation and quantification provided by the UWT-MIA approach could be used to monitor and control cell cultures in real time to detect for example the onset of fusion. Furthermore, it could be used to detect a loss of cell viability associated to a drastic change in the cellular regions texture.

5.7 Conclusions

A new multivariate imaging technique based on the Undecimated Wavelet Transform and Multivariate Image Analysis (UWT-MIA) was investigated in this paper for simultaneous extraction of spatial and spectral information. It uses a state-of-the-art continuous wavelet transform for multi-resolution texture analysis and segmentation of grey-level and multi- and hyperspectral images. Continuous wavelets have a constant resolution over all scales that provide a better frequency resolution for extracting size distribution features, as shown by Liu and Macgregor (2008). Another important advantage of the approach is that the undecimated wavelet transform yields detail and approximation sub-images which have the same size and are spatially congruent. Hence, the sub-images obtained at all scales and orientations and for each wavelength of the image can be stacked as a separate channel in a new multivariate image. Multivariate Image Analysis (MIA) can then be applied on this new image for simultaneous extraction of spectral/color and textural information in a single modelling step.

The performance of UWT-MIA was demonstrated on three sets of images. The first set consists of a RGB image where a textural pattern was replicated within the image but was given different colors and orientations. In the second set, objects of different colors and sizes where randomly distributed within the images. Grey-level images collected during live cell culture experiments were used as the third set where cell number, size, shape and orientation change as a function of culture time. In all of these cases, the UWT-MIA approach was shown to be successful at distinguishing groups of pixels belonging to regions of different colors and texture (e.g. associated with objects of different sizes, shapes, and orientations) within a single MIA model.

The UWT-MIA is therefore a very flexible and versatile technique for tackling a wide variety of textural and spectral segmentation problems. In general, continuous wavelets require tuning a greater number of parameters compared to discrete wavelets (e.g. σ, ε and k_0 for the Gabor wavelet), but they provide greater flexibility. The extraction of specific textural features for a given application can be tailored by appropriate selection of the wavelet function and its tuning parameters. In addition, all the Multivariate Image Analysis and Multivariate Image Regression (MIA/MIR) tools developed in the chemometrics literature (Prats-Montalban, de Juan et al. 2011; Duchesne, Liu et al. 2012) can be applied for segmentation in the latent variable space (i.e. score space) of the UWT-MIA model.

Finally, the method can also be cast within the multiresolutional multivariate image analysis (MR-MIA) framework proposed by Liu and MacGregor (2007). In particular, the unique model provided by UWT-MIA could serve as a sound basis for selecting which of MR-MIA I or II approach is best suited for the problem of interest. Indeed, if the loading values obtained by applying UWT-MIA in a MR-MIA I fashion are similar for all wavelengths of a multi- or hyperspectral image, this would suggest that MIA should be used first to compress the image spectral information and then followed by applying UWT on the score images to capture spatial information (i.e. use MR-MIA II instead).

5.8 Acknowledgements

The authors would like to thank Prof. J. Jay Liu for providing the basic code and insights for computing 2D Gaussian continuous wavelets. The authors would like also to acknowledge the lab of Dr. J.P. Tremblay (CRCHUL) for providing expertise/material for the culture of myoblasts, and the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) for funding.

Chapter 6. Using the undecimated wavelet transform multivariate image analysis (UWT-MIA) as an on-line tool to characterize adherent cells morphology

To be submitted to Biotechnology and Bioengineering Journal

Pierre-Marc Juneau[a], Alain Garnier[a], Carl Duchesne[a]*
[a]Department of Chemical Engineering, Pavillon Adrien-Pouliot, 1065 ave. de la Médecine, Université Laval, Québec (Québec), Canada, G1V 0A6
***Fax: +1 418-656-5993; Tel: +1 418-656-5184; E-mail:** Carl.Duchesne@gch.ulaval.ca*

Short running title: UWT-MIA to characterize cellular morphology

Index terms: UWT-MIA, phase contrast microscopy, myoblasts, morphology, culture medium, image processing, multivariate analysis.

6.1 Résumé

La morphologie cellulaire constitue un marqueur macroscopique important de la physiologie cellulaire, et la caractérisation de la forme des cellules constitue un moyen de sonder l'état d'une culture *in vitro*. À cet effet, la microscopie en contraste de phase (PCM) est un outil précieux car elle peut permettre de longues périodes d'observation en générant très peu d'artefacts de culture. Deux approches générales sont couramment utilisées pour analyser les images: la segmentation d'objets individuels et leur caractérisation par reconnaissance de patrons. La segmentation de cellules individuelles est difficile à réaliser dans les images de cellules adhérentes acquises en PCM puisque leur contour est souvent irrégulier et flou, et ce phénomène s'aggrave lorsque la culture est confluente et que les cellules se regroupent. Alternativement, les approches de reconnaissance de patrons telles que la transformée par ondelettes non-décimées combinée à l'analyse d'image multivariée (UWT-MIA), peuvent générer des caractéristiques texturales liées à la morphologie cellulaire. Un ensemble de données incorporant 200 images de calibration a été utilisé afin de calibrer des modèles de régression PLS afin d'estimer les distributions des caractéristiques morphologiques des cellules (axe majeur, axe mineur, orientation et rondeur) pour chaque image. Les modèles PLS construits ont ensuite été appliqués sur l'ensemble de la base de données incorporant 631,136 images de cultures de myoblastes acquises dans des conditions différentes. Une analyse statistique multivariée sur les données a montré que les cellules cultivées dans un milieu sans sérum étaient plus petites et plus allongées que les cellules cultivées dans un milieu supplémenté de sérum. Cette technique robuste peut donc être

utilisée sur de grands ensembles d'images de cellules vivantes (acquises en PCM) afin d'évaluer la morphologie et la cinétique de croissance cellulaire par rapport aux conditions de culture, ce qui pourrait être bénéfique pour le criblage à haut débit et les applications de contrôle automatisé de cultures cellulaires.

6.2 Abstract

Cell morphology is an important macroscopic indicator of cellular physiology, and increasing emphasis is put on cellular shape characterization as a mean of probing *in vitro* culture state. Phase contrast microscopy (PCM) is a valuable tool for observing live cells morphology over long periods of time with minimal culture artifact. Two general approaches are commonly used to analyze images: individual object segmentation and characterization by pattern recognition. Single-cell segmentation is difficult to achieve in PCM images of adherent cells since cell contour is often irregular and blurry, worsening when the culture reaches confluence and cells bundle together. Alternatively, pattern recognition approaches such as the undecimated wavelet transform multivariate image analysis (UWT-MIA), allows extracting textural features from PCM images that are correlated with cellular morphology. A Partial Least Squares (PLS) regression model built using textural features from a set of 200 ground truth images was shown to predict cellular morphological feature distributions (major axis, minor axis, orientation and roundness) accurately for each image. The PLS models were then applied on a large dataset of 631,136 images of myoblast cultures acquired under different conditions, and multivariate statistical analysis showed that the cells grown in a serum-free medium were smaller and more elongated than cells grown in a serum-supplemented medium. This robust technique can therefore be used on large datasets of PCM live-cell imaging pictures to assess cellular morphology and growth kinetics in relation to culture conditions, which could be beneficial for high-throughput screening and automated cell culture control applications.

6.3 Introduction

Changes in cellular morphology are known to be a macroscopic manifestation of different intracellular molecular processes such as the actin-polymerization pressure at the cell edge, myosin-dependent contraction of the actin network, the adhesion to an extracellular matrix and membrane tension (Ambuhl, Brepsant et al. 2012). Thus cell morphology can provide insights into the cellular state. Cellular morphology is one of the fundamental criteria used to visually classify cells into histologic types, since there is a strong relationship between the shape of the cell and its functional integrity in several specialized cells (Olson, Larson et al. 1980). Cellular morphology has been linked to growth control in normal cells, and several observations suggest that the shape of adherent cells changes during viral and oncogenic transformations (Olson, Larson et al. 1980). Cells respond to environmental signals that regulate their behaviors (cell differentiation, division and migration), and scientists as well as engineers study changes in their morphology to understand the influence of those signals (Theriault, Walker et al. 2012).

Cell morphological analysis can be performed using two different types of image analysis approaches: single-cell segmentation and pattern/texture recognition (Sommer and Gerlich 2013). The single-cell segmentation approach aims at detecting cellular regions or edges by applying a series of mathematical operations based on pixel intensity values. It can be easily implemented to analyze epifluorescence microscopy (EM) images having a good contrast between the cells and the background. Several methods are available for cell segmentation and differ essentially by the algorithms used in order to obtain a segmentation map. Once the cells are segmented, geometrical features (e.g. cell equivalent diameter, roundness, etc.) can be computed for each cell within the image. Alternatively, pattern recognition approaches (segmentation-free) are applied in cases in which segmentation of the cells is difficult to achieve because of high cell densities (cells are touching) or when dealing with complex cellular structures (Sommer and Gerlich 2013; Weber, Fernandez-Cachon et al. 2013). Single-cell segmentation in phase contrast microscopy (PCM) images remains a challenge because of the uneven illumination of the samples (Rimon and Schuldiner 2011), the bright halo and uneven contour surrounding the cells, the blurry cell-cell interfaces at high degrees of confluence and the low contrast between the background and cellular regions (Bradbury and Wan 2010; Juneau, Garnier et al. 2013). The incentive of using PCM over EM is that this technique is non-intrusive and less detrimental to the cells, since it does not require fluorescent dyes and exposure to UV light. This can be particularly useful for long-term cell culture analysis, for which

introduction of artifacts must be minimized. Since cells are difficult to segment in PCM, using pattern recognition approaches can be advantageous because of their ability to monitor cells from initial seeding to terminal high confluence, enhancing the scalability of this approach for high-throughput screening experiments (Weber, Fernandez-Cachon et al. 2013).

Pattern recognition approaches are aimed at classifying pixels or images based on the arrangements of descriptors or features (Gonzalez and Woods 2008). Image features can be seen as quantitative values extracted by image processing algorithms providing representative discriminative information (from a textural, spectral or morphological perspective) at the pixel, image region or whole image levels. Identifying the most useful features for a given pattern recognition problem is not straightforward, as feature selection is application dependent (Rodenacker and Bengtsson 2003). Textural features are of particular interest because they consider both the spatial distribution of the grey values and the overall grey values (Rodenacker and Bengtsson 2003). Several pattern recognition applications for PCM images of biological samples, relying on textural or related features, are reported in the literature. For example, grey level co-occurrence matrix (GLCM) and histogram related features were used to classify pixels related to either background, living or necrotic cells in epithelial cell cultures (Malpica, Santos et al. 2003); local binary patterns, Haralick statistics and threshold adjacency statistics features were used to distinguish (via a support vector machine classifier or SVM) human cardiomyocytes induced pluripotent stem cells from human embryonic stem cells (Paci, Nanni et al. 2011); Fourier descriptors were used to characterize differentiated and non-differentiated neural stem cells (Xiang and Datian 2009); Laplacian differential operator features were used to detect stem cells blobs by filtering the image with a series of Gaussian filters (Laplacian-of-Gaussian or LOG) (Huiming, Xiaobo et al. 2009), and shape-based and appearance-based features were used to classify fibroblasts morphological state via an Adaboost algorithm (Theriault, Walker et al. 2012). Most articles used textural and other features to distinguish cellular type, but few tried to find robust image features to study the impact of environmental conditions on cellular morphology.

Textural analysis methods available for pattern recognition belong to four categories: statistical methods (e.g. GLCM), geometrical methods (e.g. Voronoi tessellation), model based methods (e.g. random field models and fractals) and signal processing methods (e.g. wavelets) (Tuceryan and Jain 1999). Signal processing methods relying on wavelet

transforms are preferred to the Fourier transform since they consider both signal frequencies and spatial information in an image (Liu 2004). Althought most industrial applications use the discrete wavelet transform (DWT) (Tessier 2006; Gosselin 2010), the continuous undecimated wavelet transform (UWT) is increasingly used because it provides better spatial and frequency resolution (Liu and MacGregor 2008; Liu, Gui et al. 2011). The use of wavelets to analyze microscopic cellular images is still scarcely reported in the literature, as they are mostly used in EM for image denoising (Olivo-Marin 2002), for segmenting bright spots associated to cells (Padfield, Rittscher et al. 2011) or for generating textural features related to cellular state (e.g. expression of Htt-Q15/ Htt-Q138 in neurites) and to cell morphology (Wu, Schulte et al. 2010). Several PCM applications using wavelet texture analysis are available in the literature for example to assess the quality of oocytes cells for assisted insemination (Caponetti, Castellano et al. 2009) or to track suspended cells via Haar-like features (Chao-Hui, Sankaran et al. 2012). However, in most cases, PCM applications using wavelet features focus on round and isotropic cells and are not adapted for characterizing highly anisotropic adherent cells. There is still a need to develop a robust pattern recognition approach based on wavelet transform features that would be able to relate culture conditions to cellular morphology and cellular types observed in PCM.

In this work, we propose a novel PCM image analysis method based on the extraction of robust textural features computed using the undecimated wavelet transform multivariate image analysis (UWT-MIA) approach (Juneau, Garnier et al. 2015). The features extracted by UWT-MIA were found to be correlated with cellular morphological descriptors (major axis, minor axis, roundness and orientation distributions) computed from PCM images without performing cell segmentation. As an application example, these are used to distinguish the morphology of human myoblasts proliferated either in a serum-supplemented medium (SSM) or a serum-free medium (SFM). The proposed approach can be useful for online adherent cell culture control and also for high-throughput screening experiments relying on label-free microscopic methods.

6.4 Materials and methods

Human myoblasts from a primary cell line (kindly provided by Dr. J.P. Tremblay, CRCHUL) were cultivated over several passages in SSM (MCDB120 + 15% FBS) and in SFM (Parent 2009) in 6-well plates. Both media used MCDB120 as a base medium, which is an optimized nutrient medium for human muscle stem cells (Ham and Blau 1988). Inoculation

density was between 5,000 to 20,000 cells/mL, with a working volume of 2 mL/well. Growth medium was changed every 2-3 days. PCM images were captured with an IX-81 microscope (Olympus™) equipped with an automated stage and a Qimaging® monochrome camera at 20X. The resolution of the images is 0.36 µm/pixel. An incubator was mounted on the microscope stage to maintain the culture plate at 5% CO_2 and 37.5 ºC. The experimental set-up was controlled using the Metamorph® software (Molecular Devices™), and the focus was adjusted for each frame using the autofocus utility. Every 3 hours, 121 images per well were captured (one acquisition) according to an 11 x 11 rectangular grid, 0.4 mm apart. A total of 631,136 images were acquired over 23 experiments (138 wells), from seeding to confluence. Image quality was assessed manually for each acquisition (overall quality for each acquisition of 121 images).

Image analysis using the UWT-MIA approach was implemented within the Matlab® environment. The Image Processing Toolbox® was also used to develop the custom scripts. Multivariate data analysis was performed using either the PLS Toolbox® (Matlab®) or SIMCA-P® (Umetrics™). A set of 200 ground truth images in which individual cells were segmented manually in a previous study (Juneau, Garnier et al. 2013) was used as a calibration dataset.

6.5 Image processing and data analysis

Several steps are required to estimate cellular morphological features for a whole database (illustrated in Figure 6-1):

1. Performing UWT-MIA on the calibration dataset;
2. Calibrating PLS models and mask parameters to estimate cellular morphological features;
3. Computing cellular morphological features for the whole database.

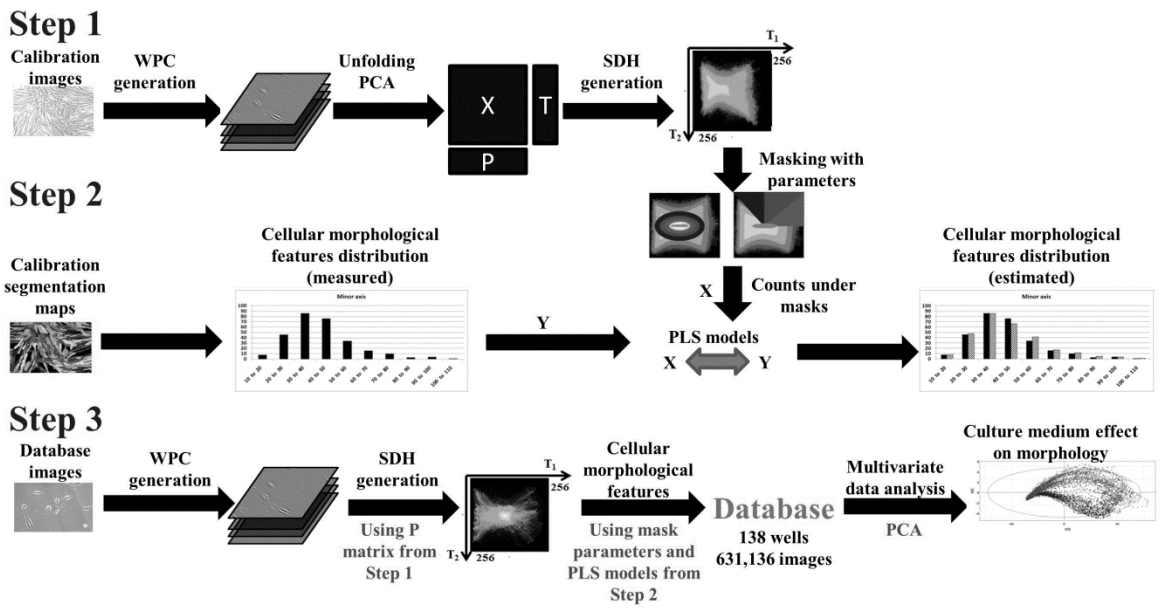


Figure 6-1 : Major steps involved in the proposed pattern recognition approach to estimate and analyze cellular morphological features. Using wavelet planes cubes (WPCs), it is possible to generate score density histograms (SDHs) by performing a principal component analysis (PCA) and to compute morphological features via Partial Least Squares (PLS) models.

In Step 1, the images in the calibration dataset (200 images) are submitted to the UWT-MIA: wavelet planes cubes (WPCs) are computed (one per image), the wavelet coefficients are unfolded and a principal component analysis (PCA) is performed. $\mathbf{T}_1/\mathbf{T}_2$ score density histograms (SDHs) are computed for each image of the calibration dataset.

In Step 2, the calibration segmentation maps (associated to the 200 calibration images) generated manually are used to compute measured cellular morphological features (serving as a **Y** matrix). The SDHs from Step 1 are used to compute pixel counts under defined masks (designed using specific parameters) in order to compute UWT-MIA textural features (serving as an **X** matrix). With both the **X** and **Y** matrices, optimal PLS regression models are identified (along optimal mask parameters) via grid search in order to estimate cellular morphological features.

In Step 3, each image in the complete database is processed, and for each a WPC is computed. WPCs are projected into the latent space defined in Step 1 using the loadings matrix **P** to compute SDHs. Using the PLS models regression coefficients and mask

parameters determined from Step 2, estimates of the cellular morphological features are computed for each image in the database. The data generated is used to correlate morphological traits to culture conditions or cell types by means of multivariate analysis.

6.5.1 UWT-MIA performed on the calibration dataset

The UWT-MIA (Juneau, Garnier et al. 2015) mainly consists of computing a Gabor wavelet plane transformation at different scales and orientations on the image set, followed by performing a Principal Component Analysis (PCA) (Wold, Esbensen et al. 1987) on the stack of wavelet planes. This allows capturing textural and spectral information from each image at the pixel level. Pixels associated with similar textural patterns within the images will cluster together in the PCA scores space where the different textural regions should be discriminated. The way to apply MIA on several WPCs is to use a kernel algorithm (Geladi and Grahn 1996), which involves computing the variance-covariance matrix for all the WPCs followed by a singular-value decomposition (SVD) to estimate a common PCA latent variable space based on all the images of the set.

The UWT-MIA was performed on the 200 calibration images for which PCA loadings and the global $\mathbf{T}_1/\mathbf{T}_2$ score density histogram are presented in Figure 6-2 and Figure 6-3, respectively. The first component (related to pixels intensities) explains 92.2% of the variance in the WPCs as the second component (related to the texture inside the images) explains 2.0% of the variance. Each WPC incorporates 400 Gabor wavelet planes that were constructed using scales s =1 to 50 (by increments of 1) and angles θ = 0 to 7π/4 radians (by increments of π/4) using the following Gabor wavelet parameters: σ = 0.5, ε = 1 and k_0 = 5. These parameters were adjusted by trial-and-error using a few images to roughly match the shape of a myoblast cell imaged in PCM. This approach was adopted because there are no formal guidelines in the literature to select the type of wavelet that would be appropriate for a given image processing problem. To generate the score density histogram $\mathbf{T}_1/\mathbf{T}_2$, the scores were rescaled between 0 and 255.

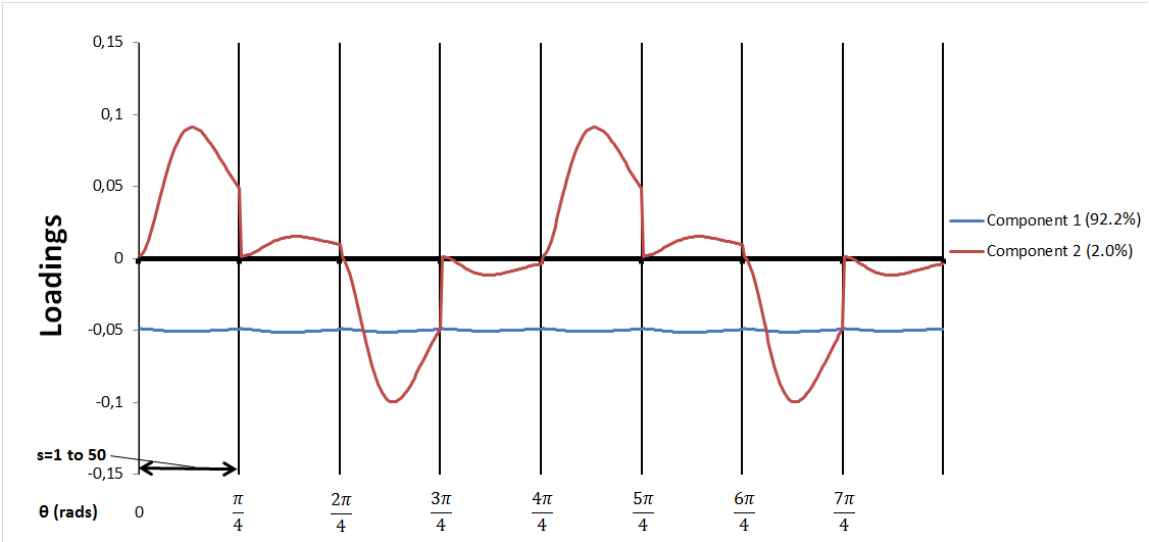


Figure 6-2 : Loadings for PCA performed on the WPCs of the 200 calibration images. The blue and red lines correspond to the first and second components, respectively.

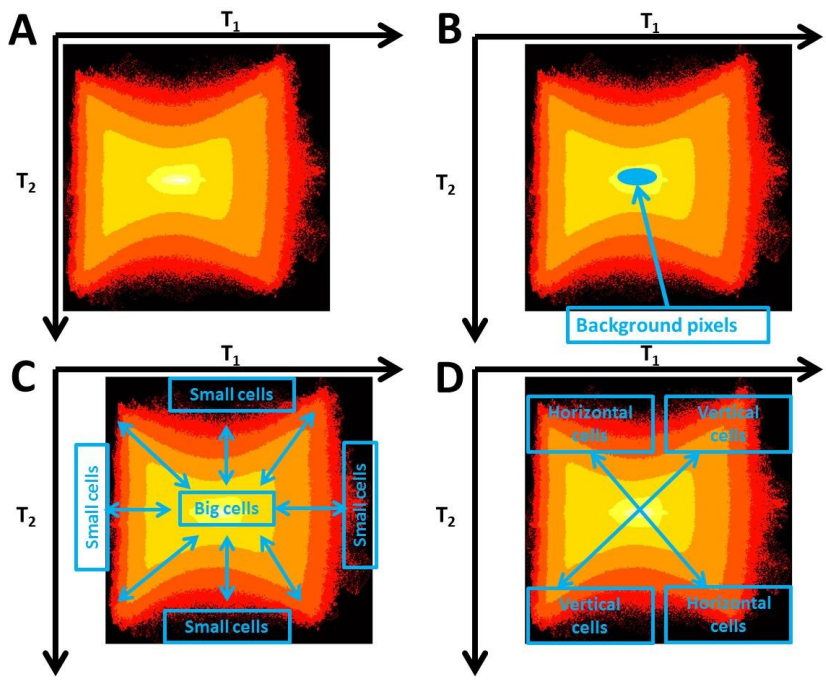


Figure 6-3 : A) T_1/T_2 score density histogram of the global PCA model built from the 200 calibration images. The color scale (from black to white) indicates the density of pixels having a certain combination of T_1/T_2 values. Pixels associated with B) background regions C) cell of different sizes, and D) orientations.

Figure 6-2 provides the loadings (and the relative importance) associated to each wavelet plane in a given latent component: the loadings associated to a specific orientation θ are enclosed within vertical lines. Between two vertical lines it is possible to see the relative influence of scale in the textural information captured (for a constant orientation θ). As shown in Figure 6-2, the variance captured by the first component is related with changes in grey level intensities across the image since it has uniform negative loadings (all channels have the same sign) and no relationship with scale or orientation. The second component captures textural information related with the size and the orientation of the cells in the images because the loadings are positive for vertical patterns (θ=0 and 4π/4) and negative for horizontal patterns (θ=2π/4 and 6π/4), and also vary as a function of scale. Figure 6-3 shows the raw $\mathbf{T}_1/\mathbf{T}_2$ global score density histogram (A) as well as masks identifying the regions where image pixels are associated with background (B), cells of different sizes (C), and orientations (D). Note that pixels associated to large cells (low frequency signal) usually end-up around the background pixels region (very low frequency signal). The traditional way of extracting features using MIA is to mask a region in the features space (score density histogram), then count the number of pixels falling under the mask, and finally overlay these pixels in the original image to interpret the information captured by MIA (Duchesne, Liu et al. 2012). The masks can be determined manually by trial-and-error or can be determined by an automated supervised method (Liu, Bharati et al. 2005). In this work UWT-MIA features were calculated by using a series of masks segmenting the regions within the score density histogram $\mathbf{T}_1/\mathbf{T}_2$ related with the orientation and the size distribution of the cellular population.

The main output of the UWT-MIA are the individual $\mathbf{T}_1/\mathbf{T}_2$ score density histograms (SDH, one per image), which can be used to provide textural features via the masking procedure. The SDHs contain information about groups of pixels belonging to certain textural regions associated to cells of different shapes, sizes and orientations. Figure 6-4 provides several images with their associated SDH for low (A) or high (B) cell confluence, for round cells (C) and for elongated cells oriented vertically (D) or horizontally (E). Examples of masking/overlays in the $\mathbf{T}_1/\mathbf{T}_2$ global score density histogram are provided in Figure 6-5 for background pixels (A), for big or round cells (B) and for small or elongated cells (C).

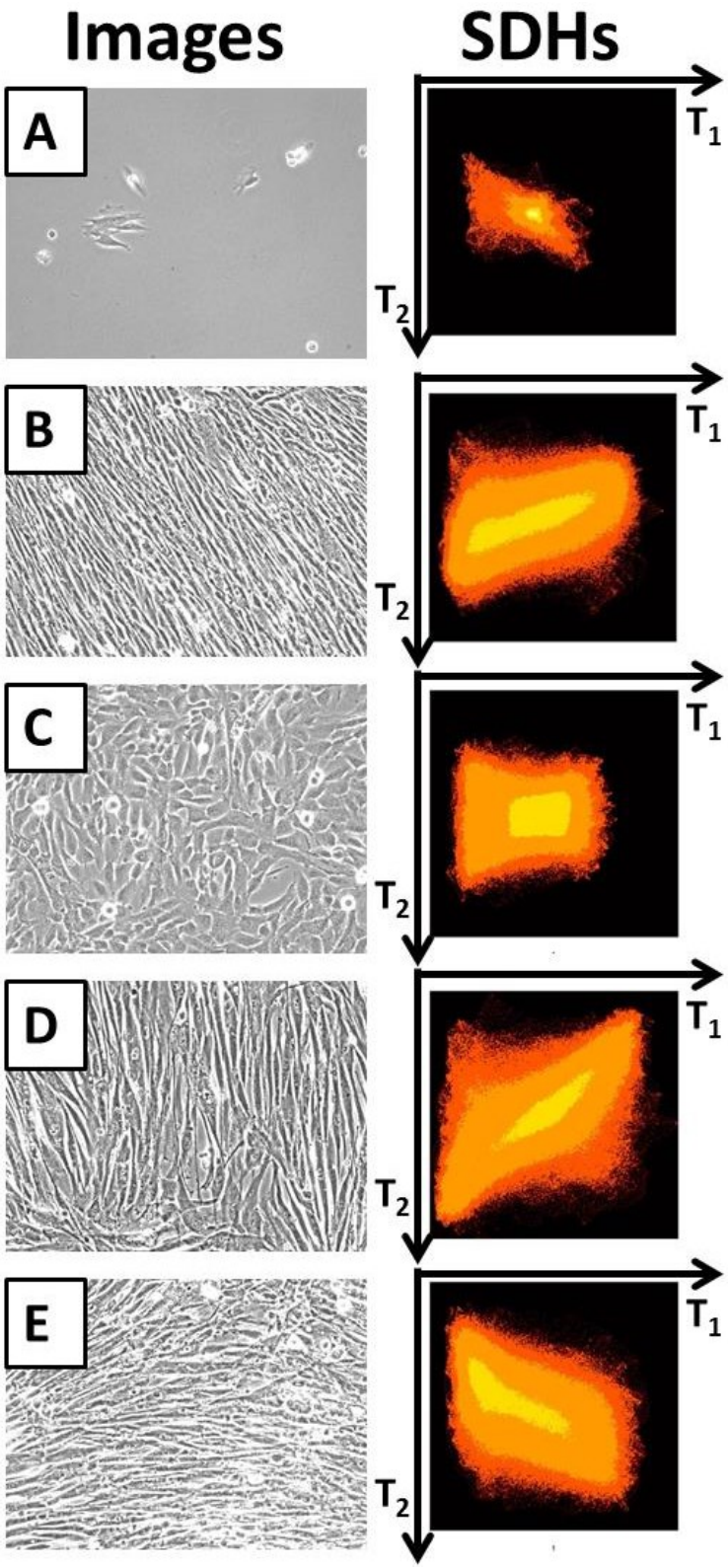


Figure 6-4 : Images and their SDH for low (A) or high (B) cell confluence, for round cells (C) and for elongated cells oriented vertically (D) or horizontally (E).

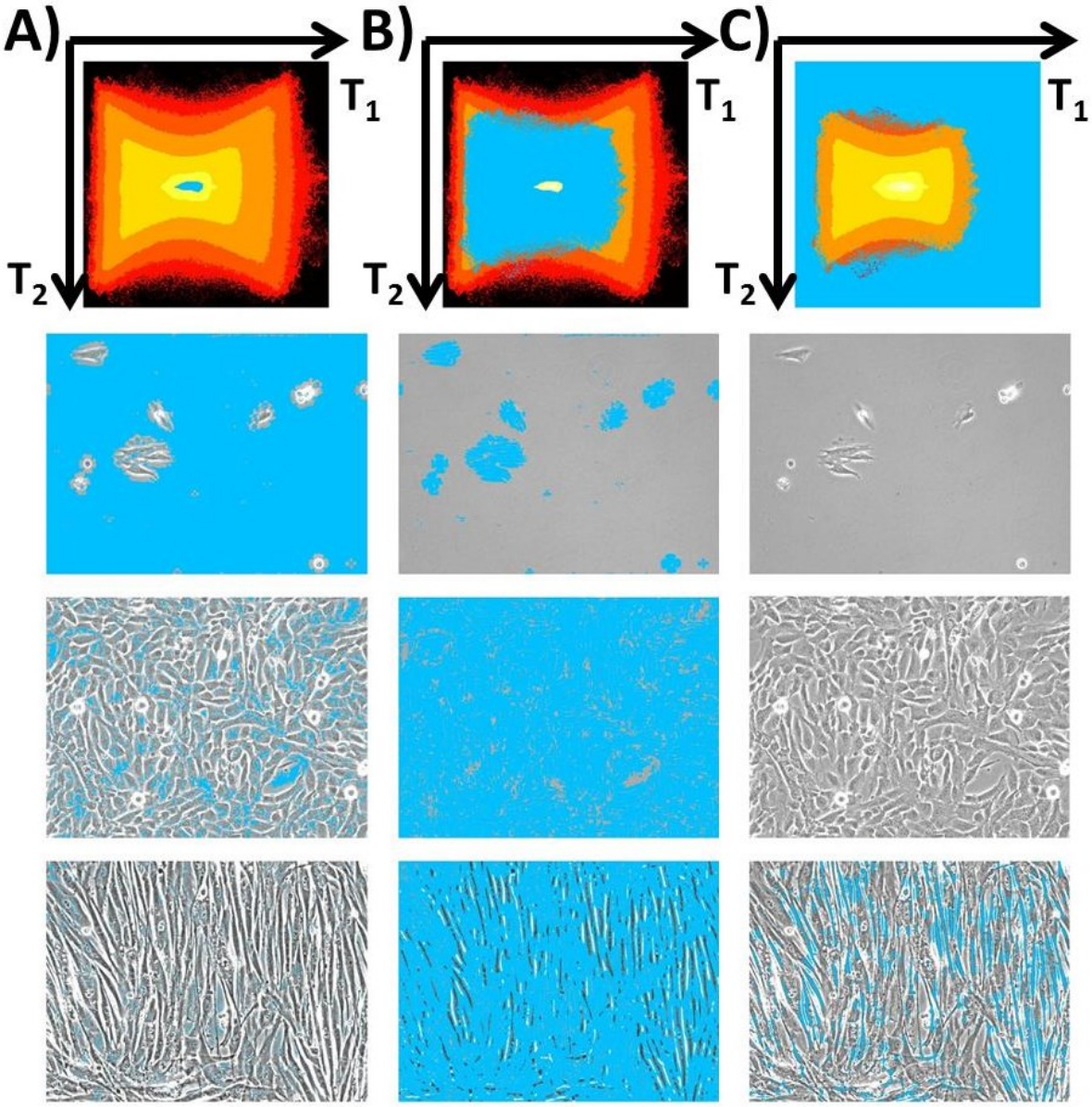


Figure 6-5 : Illustration of the masking procedure in the T_1/T_2 score density histograms. Three different masks are applied (cyan color) on pixel regions associated with the background region (A), round or large cells (B) and small or elongated cells (C). The identified pixels are then overlaid onto three PCM images.

6.5.2 Extraction of relevant image features for cellular morphological characterization

The calibration dataset (200 hand-segmented ground truth images)(Juneau, Garnier et al. 2013) was processed by UWT-MIA and was used to calibrate Partial Least Squares (PLS) (Wold, Sjostrom et al. 2001) models in order to predict cellular morphological features such as the distribution of major and minor axes, the cell roundness and orientation. The roundness of a cell is defined as the ratio of its minor axis over its major axis length. A circular cell has a roundness value around 1 whereas a very elongated cell has a value tending towards zero.

The cell major axis, minor axis, roundness and orientation distributions are generated by computing the number of cells falling in specific classes from the manual segmentation map generated by the experimenter (**Y** variables). The matrix of predictors (**X**) used for building the PLS models consists of the UWT-MIA features obtained by counting the number of pixels falling under different masks (areas of the individual score density histogram $\mathbf{T}_1/\mathbf{T}_2$), extracting information about each cellular feature of interest. The masks are shown in Figure 6-6. To capture the features related with cell sizes (cell major axis, minor axis and roundness distributions), a series of concentric elliptical masks around the background pixels region (manually drawn) is used. The major axis length of these masks is parameterized by a_n and its minor axis by $r*a_n$. The width of the annular region formed by two concentric masks is Δa (Figure 6-6A). To generate features for the cell orientation distributions, the score density histograms are divided into a series of triangular masks. The size of these mask is parameterized by angle $\Delta\Omega$. Note that the background mask used for estimating the cells orientation distribution (Figure 6-6B) is larger than the manually drawn background mask used for the other cell size distributions (Figure 6-6A) in order to focus on high frequency information related with cellular orientation.

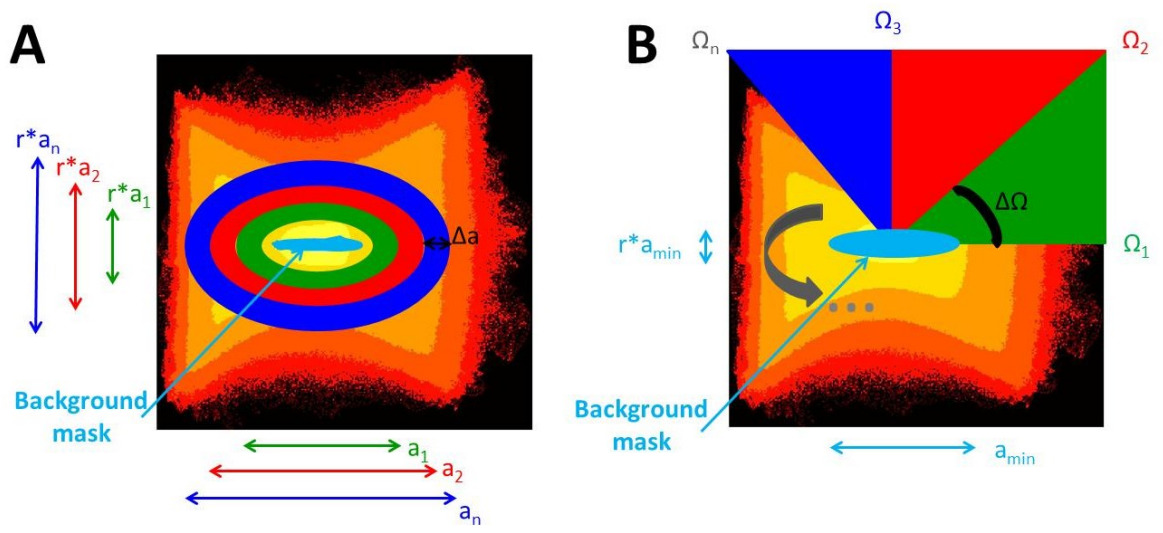


Figure 6-6 : Parameters associated with the series of masks used for estimating A) cell size related distributions and B) cell orientation distribution.

The 200 ground truth calibration images were divided into two groups (2/3 calibration, 1/3 validation) in a random fashion in order to have two representative groups (from a degree of confluence perspective). The masks parameters and the number of PLS components (N_{comp}) were determined by grid search to maximize the predictive power of the model in calibration and validation (R^2_{CAL} and R^2_{VAL}) for the total number of cells as well as the number of cells in each individual morphological feature class. Table 6-1 provides the distribution specifications for **Y** variables (cellular morphological features), the optimal mask parameters for **X** variables (to generate UWT-MIA features) and the statistics quantifying the performance of the PLS models. Note that the PLS analysis was performed on each **Y** matrix associated to each morphological features distribution as a whole (PLS-2 method). The PLS models provided a good predictive performance for most of the cellular feature distributions (R^2_{CAL} and R^2_{VAL} values above 90% for most morphological features). Note that the 200 ground truth calibration images include a variety of cell shapes, orientations and densities that were grown in two different culture media (SSM and SFM).

Table 6-1 : Optimal mask parameters and PLS models performance for each cellular morphological features distribution estimated using UWT-MIA features.

	Major axis distribution	Minor axis distribution	Roundness distribution	Orientation distribution
Distribution Specifications (for Y)				
Number of classes	22	10	4	3
Smallest class	50 - 60 pixels	10 - 20 pixels	0 - 0.25	0°
Largest class	260 - 270 pixels	100 - 110 pixels	0.75 - 1.00	90°
Classes resolution	10 pixels	10 pixels	0.25	45°
Optimal Mask Parameters (for X)				
a_{min} (pixels)	62,59	62,59	62,59	126,59
a_{max} (pixels)	256	256	256	-
Ω_{min} (rads)	-	-	-	0
Ω_{max} (rads)	-	-	-	$29\pi/16$
r	0,35	0,15	0,3	0,15
Δa (pixels)	1	1	1	-
$\Delta\Omega$ (rads)	-	-	-	$3\pi/16$
PLS Models Performance				
N_{comp}	11	7	7	9
R^2_{CAL} total number of cells	0,954	0,897	0,940	0,938
R^2_{VAL} total number of cells	0,901	0,922	0,939	0,939
Max R^2_{CAL} individual classes	0,934	0,887	0,947	0,960
Min R^2_{CAL} individual classes	0,725	0,436	0,673	0,868
Avg R^2_{CAL} individual classes	0,837	0,737	0,834	0,915
Max R^2_{VAL} individual classes	0,821	0,899	0,943	0,945
Min R^2_{VAL} individual classes	0,364	0,211	0,682	0,770
Avg R^2_{VAL} individual classes	0,613	0,579	0,777	0,834

Examples of the performance of PLS models to estimate cellular morphological features distributions are provided in Figure 6-7 using two such PCM images (from the calibration set). Small differences can be seen between the ground truth data and the PLS predictions for individual morphological features, but the overall distributions are well represented. There is a total of 39 morphological features computed for each image from PLS models (i.e. bins of the discrete distributions of major and minor axes, roundness and orientations).

The PLS models built using ground truth calibration images were then applied on the entire image dataset to assess the effect of culture medium on myoblast morphology. Also, for each image, the cell-covered surface ratio (CCSR) was computed (Juneau, Garnier et Duchesne (2013)), defined as the ratio of the number of pixels belonging to cells over the total number of pixels in the image (measure of confluence). An additional multivariate analysis was then performed on the estimated morphological features distributions to highlight the main relationship between myoblast morphology and culture medium (as an application example).

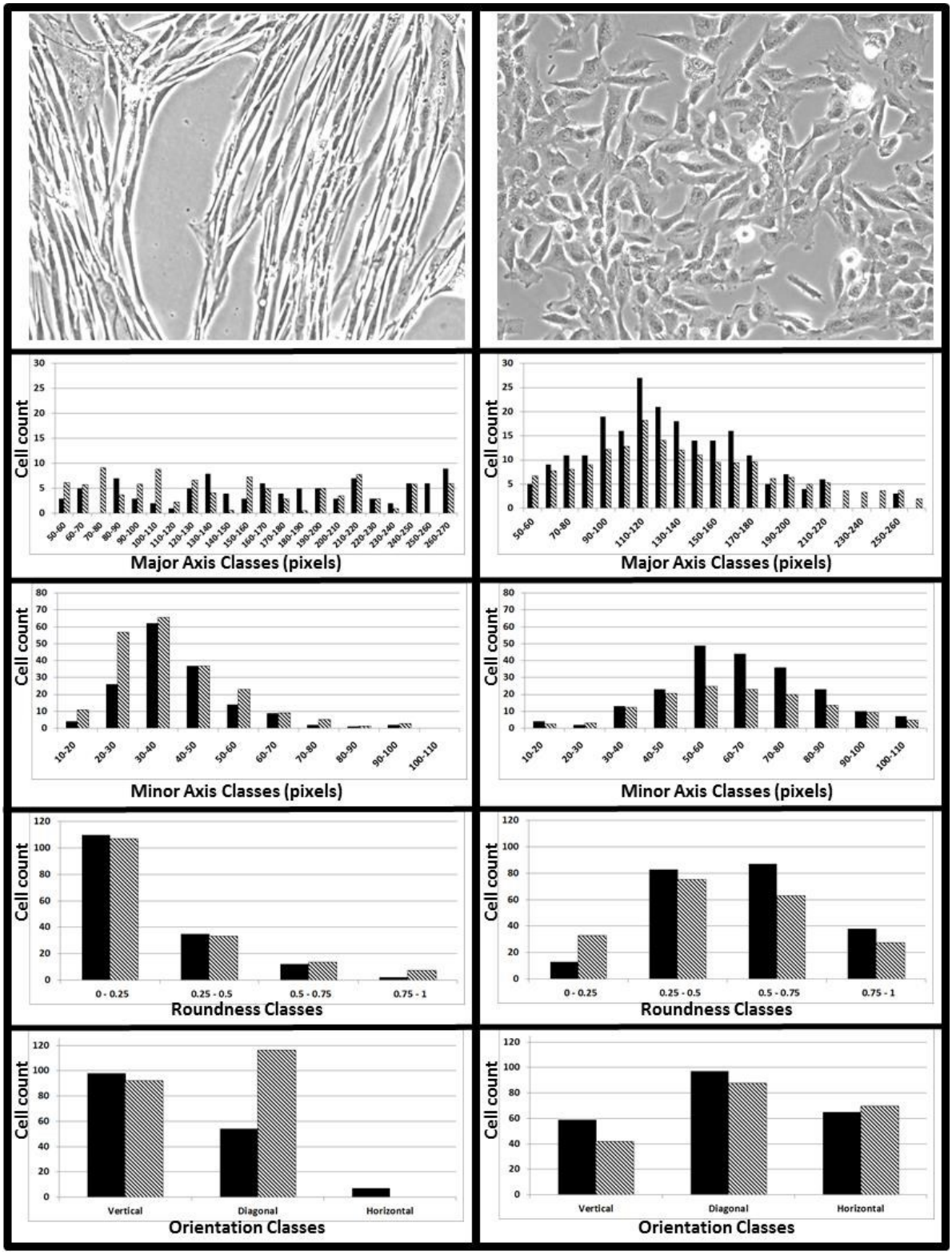


Figure 6-7 : Performance of PLS models for estimating cellular morphological features distributions illustrated on two calibration images. Black bars are manual cell counts (ground truth data) and hatched bars are cell counts estimated by the PLS models.

6.5.3 Estimation of cellular morphological features for the entire PCM image database

The PLS models were then applied to the whole PCM images dataset (631,136 pictures) and morphological cellular features were computed for each image. The PLS predictions results were used to assess the evolution of the morphological features through culture time and to discriminate cells grown in SSM from those in SFM.

6.5.3.1 Evolution of morphological cellular features through culture time

The CCSRs and the morphological features were computed for each image in each of the 138 wells. For each well, it is thus possible to plot the evolution of these features through culture time, from seeding to confluence. To illustrate this concept, results from two typical wells are shown in Figure 6-8, one well for myoblasts grown in SSM and another well for myoblasts grown in SFM (both at low passage). For each acquisition time, the average CCSR over 121 images is computed (11x11 grid). Also, an average cell count (cells/image) was obtained by summing the estimated cell counts (from the PLS models) in the individual bins for the minor axis distribution (see Figure 6-8). Note that a similar computation to have estimated cell counts can be performed also on either the major axis, the roundness or the orientation distributions. The evolution of the average cell counts and the average CCSR for both wells are shown in Figure 6-8. As can be seen, there is a good correlation between the average CCSR curves and the average cell counts, both media supporting similar growth kinetics (confirming the results presented in Juneau, Garnier and Duchesne (2013)).

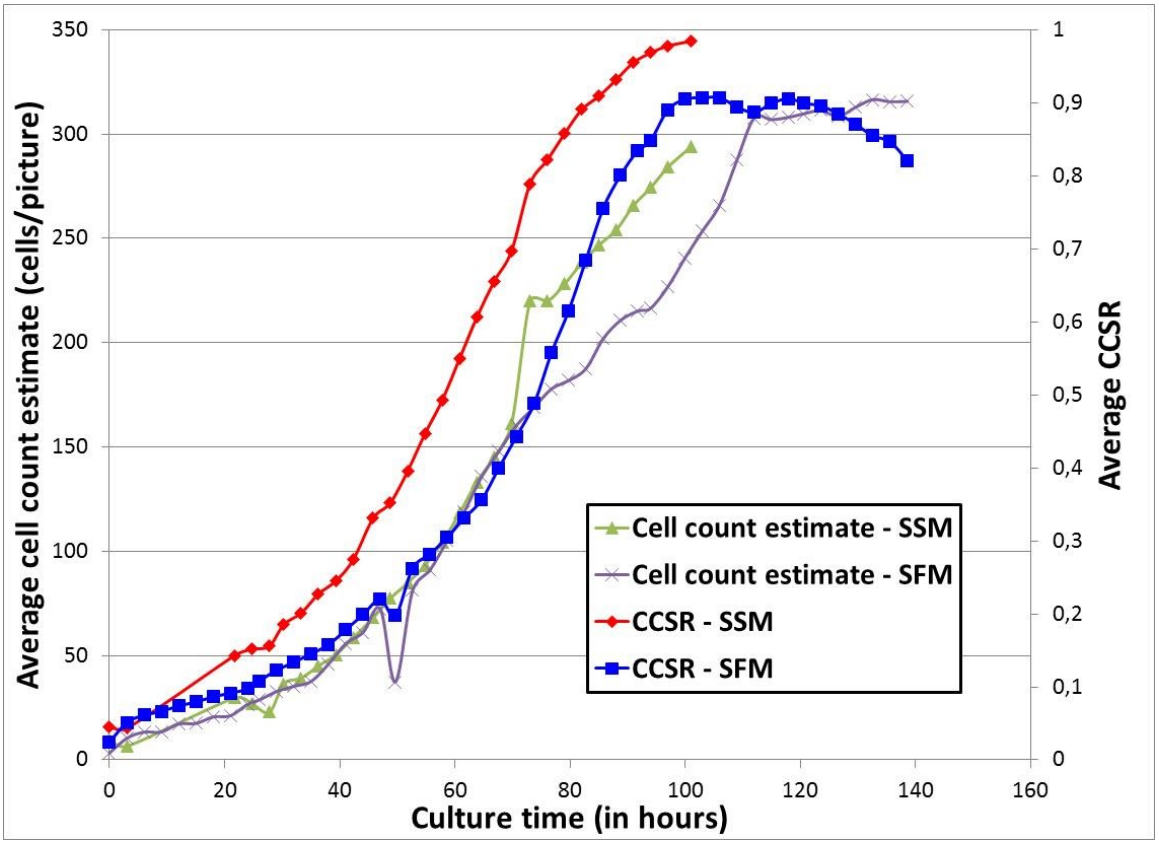


Figure 6-8 : Average CCSR and average cell count curves (sum of individual minor axis bins) for a typical SSM and SFM well.

Using morphological features distributions, the average minor and major axis lengths, the average cellular orientation and the average cellular roundness for each acquisition (over 121 images) were computed. For the same two typical wells, these metrics are shown in Figure 6-9.

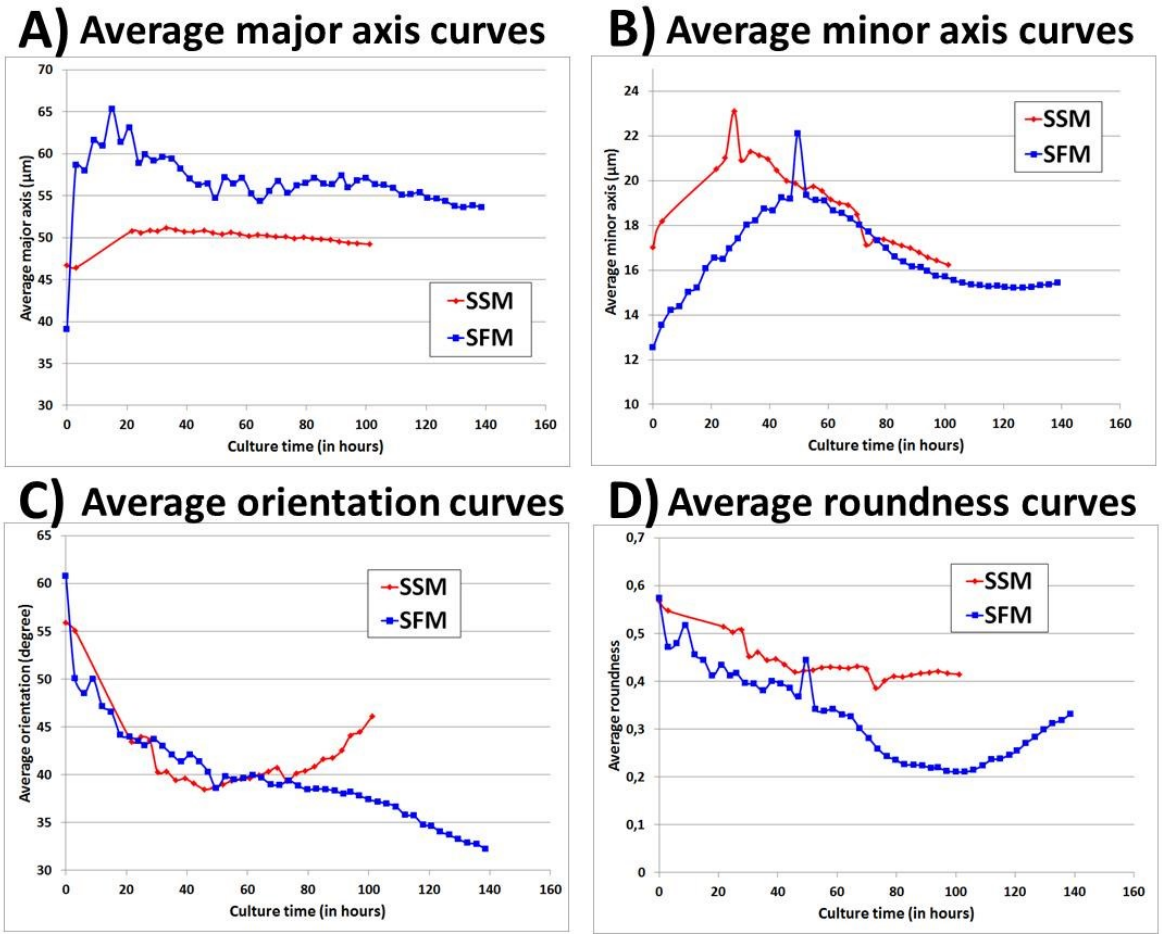


Figure 6-9 : Average major axis (A), minor axis (B), orientation (C) and roundness (D) curves for a typical SSM and SFM well. SSM curves are in red and SFM curves are in blue.

As can be seen in Figure 6-9, cells grown in SFM are longer (higher average major axis), thinner (lower minor axis) and less round (lower average roundness) than when they are grown in SSM. Furthermore, the average cell orientation changes for both wells through time, as cells bundle together (as CCSR increases).

The fetal bovine serum (FBS) lot used in SSM can also influence growth kinetics and cellular morphology. Figure 6-10 shows that with FBS lot #2 (FBS2), myoblasts growth is much slower (lower CCSR and cell count) and produce smaller cells in terms of major axis and minor axis (on average) than with FBS lot #1 (FBS1).

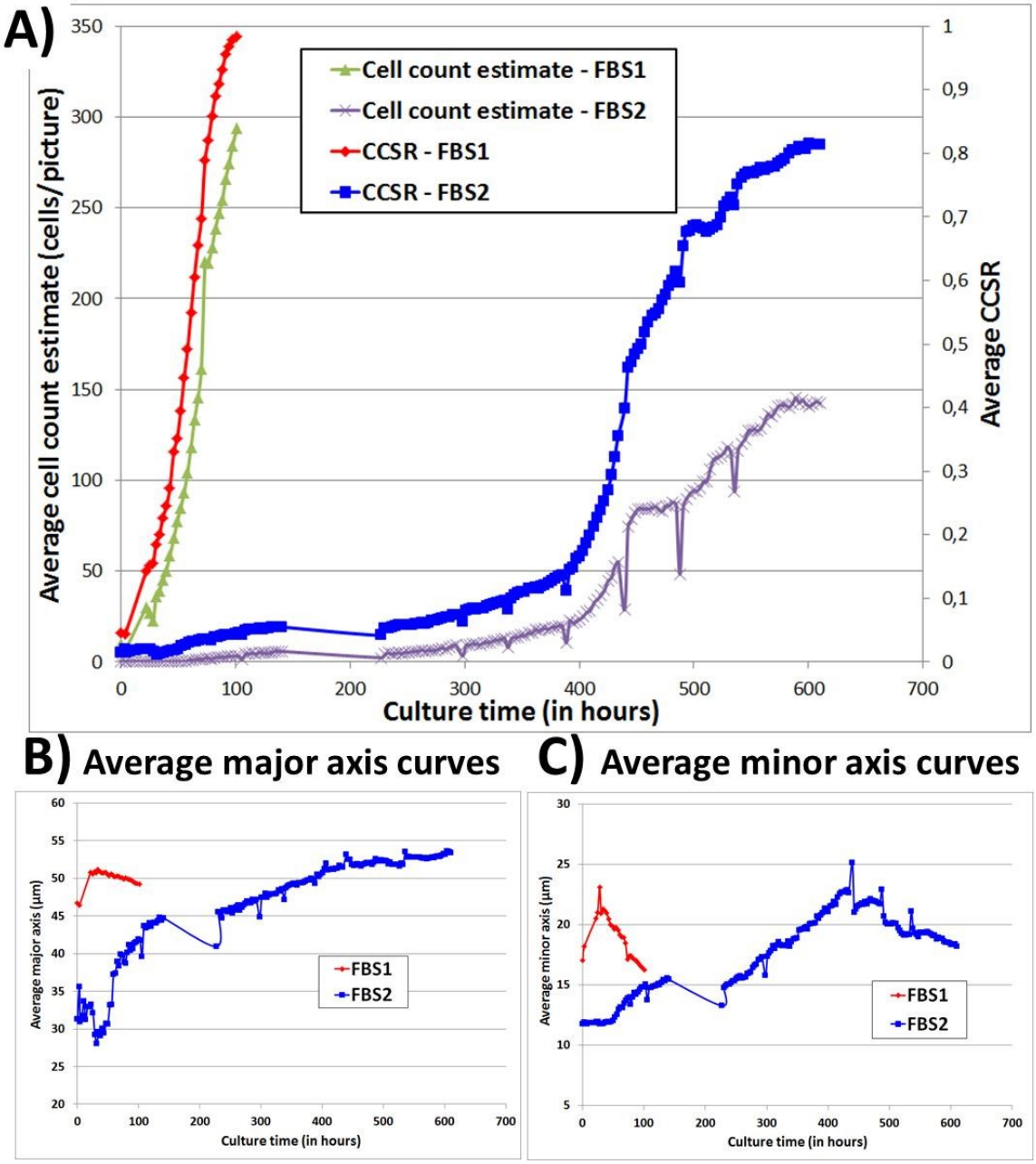


Figure 6-10 : Average CCSR and average cell count curves (sum of individual minor axis bins) (A), major axis (B) and minor axis (C) for two SSM wells with FBS1 (red) and FBS2 (blue).

6.5.3.2 Multivariate analysis to assess the effect of culture medium on cellular morphology

The major axis, minor axis, roundness and orientation distributions for each image of the database were computed using the PLS models, generating 39 morphological features. For each image, an estimate of the CCSR was also computed. Morphological features and

the CCSR for each acquisition (121 images x 40 features) were averaged and used in further analyses. After bad quality images were removed by the experimenter (i.e. mainly because of wrong focus), a total of 5,085 data points (corresponding to a total of 615,285 PCM images) were analyzed.

A new PCA model was built within SIMCA-P® on all the data points to capture most of the variance in the cellular morphological features space. Using 7 components selected automatically by cross-validation, as much as 99.7% of the variance was captured. Components 1 and 2 alone captured 86.9% and 9.6% of the variance, respectively. The two $\mathbf{T_1}/\mathbf{T_2}$ scores scatter plots in Figure 6-11 shows the information captured by each component using color maps.

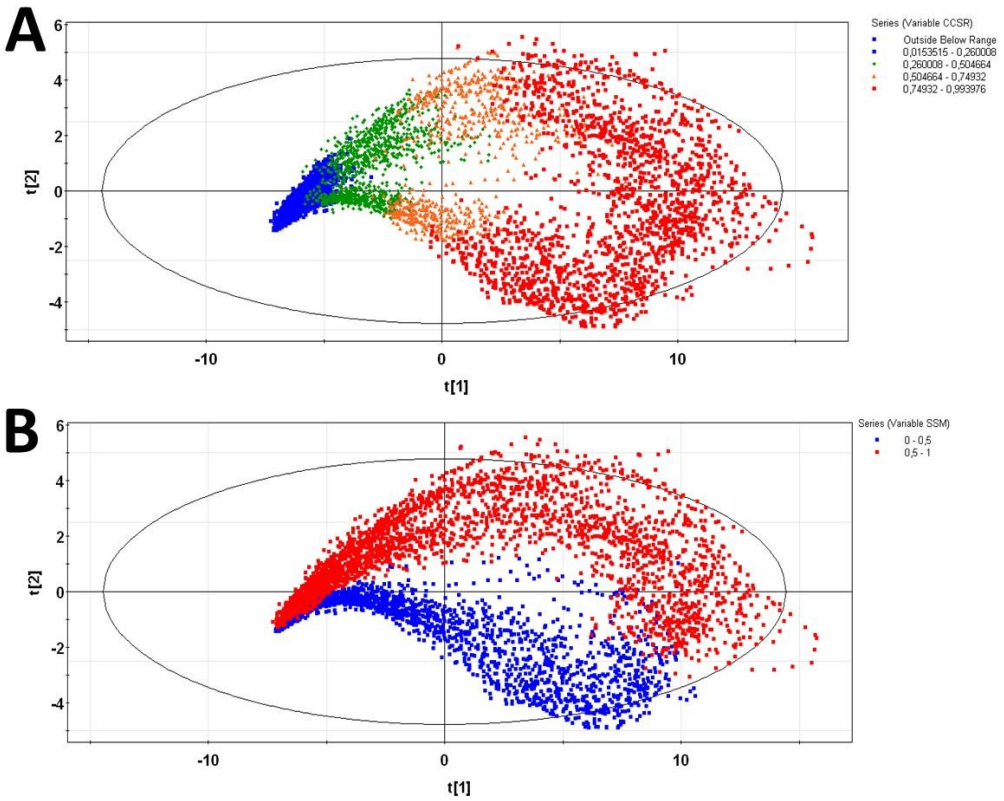


Figure 6-11 : : T_1/T_2 scores scatter plots of the PCA model built on the morphological features extracted on the entire PCM image dataset. A) Points are colored according to CCSR, from low (blue) to high (red) values. B) Points colored according to the culture medium used, red (SSM) and blue (SFM).

The first principal component (**T**$_1$ scores) clearly captures information related with CCSR (increase of confluence through time), as low ratios (blue) clearly appear on the left of the graph and high ratios (red) on the right (Figure 6-11A). In Figure 6-11B, at low CCSRs (at low degrees of confluence, at the beginning of the cell cultures), there is not a significant difference between cells grown in SSM (red) and cells grown in SFM. However, as the CCSR increases (from left to right), two clusters emerge and are distinguished by the culture medium used (top right is SSM in red and bottom right is SFM in blue).

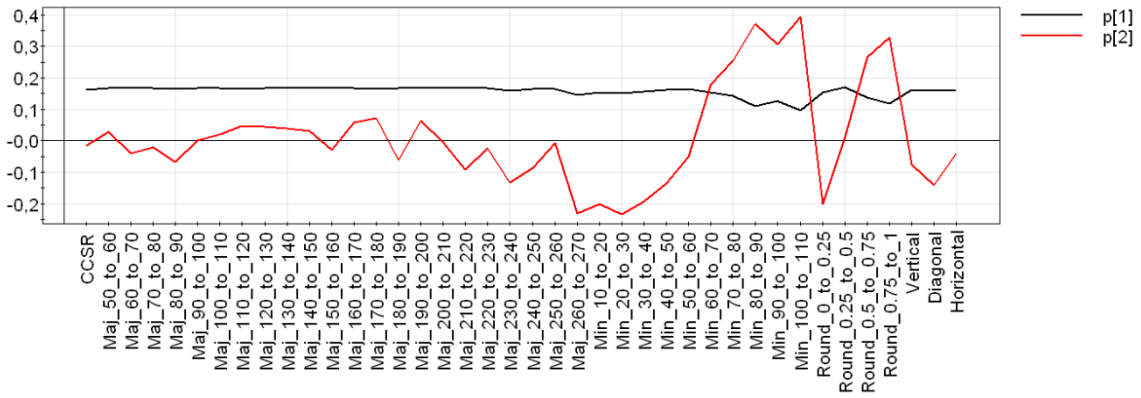


Figure 6-12 : Loadings of the first (black) and second (red) PCA components. Maj corresponds to the major axes (in pixels), Min corresponds to the minor axes (in pixels) and Round corresponds to roundness.

Figure 6-12 shows the loadings associated to the first two PCA components. It can be seen that all the morphological feature variables have similar loadings for the first component (black), which is expected since the number of cells detected in each cellular features bin is roughly proportional to the CCSR (and the degree of confluence). However, for the second component, variables associated to the minor axis and roundness distributions seem to be the most critical to distinguish cells grown in SSM from cells grown in SFM. In Figure 6-11B, blue points associated to negative scores (SFM cells) are mostly influenced by variables with negative weights (in Figure 6-12) as red points (SSM cells) associated to positive scores are influenced by variables with positive weights. These results show that cells grown in SSM are rounder (roundness variables from 0.5 to 1 have strong positive weights) and are larger than average (minor axis variables from 60 to 110 pixels have strong positive weights). In comparison, cells grown in SFM are more elongated (strong negative weights for roundness variable 0-0.25 and for major axis length

variable 260-270 pixels) and are finer (strong negative weights for minor axis variables from 10 to 60 pixels).

The main conclusions drawn from the PCA analysis are as follows. Cells grown in SFM are thinner relative to their minor axis length, with the most important contributors in the 10 to 60 pixels range (4 to 22 µm) compared to cells grown in SSM, with important contributors in the 60 to 110 pixels range (22 to 40 µm). SFM cells are also more elongated on average, with more cells that have a 260 to 270 pixels length (95 to 99 µm) and a roundness below 0.25. This is in agreement with experimental observations, in which cells grown in SFM are finer, a little smaller and more elongated than cells grown in SSM. Typical images are shown in Figure 6-13 (scale bar provided).

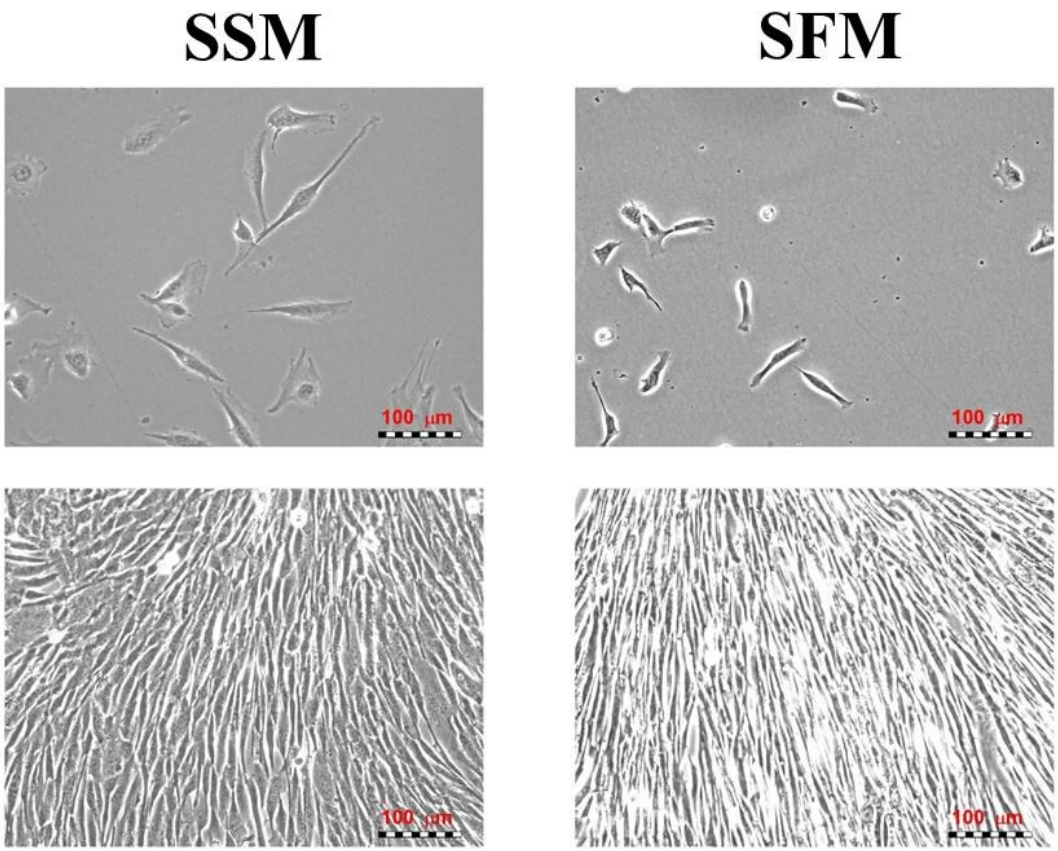


Figure 6-13 : Typical images of myoblasts grown in SSM and SFM at low and high CCSR (at confluence).

6.6 Discussion

Culture conditions may have an influence on cellular state, differentiation and morphology, and image processing techniques relating the effect of culture medium composition to morphological features can provide useful insights. For instance, the effect of serum and insulin concentrations on the differentiation of mesenchymal stem cells and lipid accumulation in adipocytes was assessed using morphological features relying on image processing (Or-Tzadikario, Sopher et al. 2010). In the case of myogenic cells, myoblasts proliferate and migrate in a random fashion on a substrate and align themselves as they reach confluence and differentiate to form myotubes (Freshney 2000). The culture medium used can affect myogenic cellular state and thus the morphology and subsequent differentiation. For example, when inducing cellular fusion, freshly isolated muscle precursor cells (MPCs) that were expanded and differentiated in the presence of rh-Wnt3A formed larger myotubes (with a greater average size) than untreated MPCs (Agley, Velloso et al. 2012). The impact of the culture medium used in the current work was already assessed qualitatively: myoblasts grown in SFM (LOBSFM) contained more granules, formed more vacuoles and were more elongated (more delicate) than myoblasts grown in SSM, showing a certain impact of the culture media on cell state. However it was shown quantitatively that the SFM was providing similar cellular growth kinetics than SSM (Parent 2009; Juneau, Garnier et al. 2013), while maintaining cellular phenotype. For both SFM and SSM, this qualitative assessment was confirmed in a quantitative manner by applying multivariate statistical methods on cellular features generated from image processing. The possibility to monitor the evolution of morphological cellular features distribution through time using PLS models was demonstrated. The time series provided in Figure 6-8, Figure 6-9 and Figure 6-10 show that these could be used as potential soft sensors for cell culture monitoring and control.

As an application example, the PCA analysis showed that at the beginning of the cell culture (CCSRs below 26%, blue points in Figure 6-11A), the cells grown in both culture media have similar morphologies. This can be explained by the fact that just before being seeded, the cells are trypsinized and suspended (become spherical), and it takes some time before they recover their functional shape relative to the culture conditions used once they adhere to a new substrate. The PCA analysis reveals the possibility to distinguish two cellular populations (SFM versus SSM) as soon as cells reach a CCSR over 25-26% (green points in Figure 6-11A). It is expected that changes in a single culture media or culture conditions through time could be detected in a similar fashion. On average, cells

need 55 ± 10 hours (at passage 5 or 6) to reach a CCSR of 25% so the effect of culture medium on the morphology of both populations becomes visible. The PCA loadings plot interpretation confirmed in a quantitative manner that cells grown in SFM are more elongated and smaller (roundness distribution leaning towards lower values, minor axis distribution leaning towards lower values). The method developed in the current work together with culture experiments suggest that it may be possible to track the impact of changes in culture media on cellular morphology using PCM images. Assuming that morphological changes can be linked to the cell functionality, the proposed approach may be used for cell culture monitoring in a non-intrusive manner. Once calibrated, the PLS models built can be used on a significant quantity of data. Multivariate analysis performed on the morphological features can be used to distinguish cell populations grown in different culture conditions (if culture conditions have an impact on cellular shape or growth kinetics).

Since UWT-MIA features rely on pattern recognition, it avoids issues associated to single-cell segmentation methods. The accurate segmentation and measurement of morphological characteristics of adherent cells with non-uniform shapes (such as cells of the myogenic lineage) is difficult (Agley, Velloso et al. 2012). Myoblasts usually generate pseudopods, which are transient protrusions of the cellular membrane used for locomotion and phagocytosis which are important to detect correctly via image processing when performing mobility and morphological studies based on single-cell segmentation (Zimmer, Labruyere et al. 2002). Because pseudopods are usually partly or entirely localized outside of the focal plane, they appear at a much lower contrast than most of the cell membrane in PCM, and thus tend to be treated by snakes (edge detection method for segmentation) as occlusions and are ignored (Zimmer, Labruyere et al. 2002). The UWT-MIA method offers an alternative solution to this problem since it uses a pattern recognition approach relying on wavelets and textural analysis.

Signal processing approaches are appropriate tools for cell detection in PCM images because cell-associated pixels are related to higher frequency signals in comparison to background pixels (Zhaozheng and Kanade 2011). Other multiscale transform-based texture analysis methods are also proposed in the literature to generate features that can be fed to a classifier for pattern recognition purposes. For instance, in the Scale Invariant Feature Transform (SIFT), a Difference-of-Gaussians (DoG) approach is used in which images are convolved with Gaussian isotropic filters of varying standard deviations (σ)

over several scales, and for which difference images between two consecutive Gaussian-blurred images (differing by their σ) are computed in order to highlight key features (Lowe 1999; Lowe 2004; Jiang, Crookes et al. 2010). Key points (features) are identified by finding local minima or maxima in the difference images (Lowe 2004). SIFT features have the advantage of being robust to object orientation, scale, illumination and don't require pre-segmentation (Lowe 1999). SIFT features were used to detect mitosis of stem cells in PCM, since they are invariant to image scale and rotation, robust to illumination fluctuations, noise, partial occlusion, and minor changes in viewpoint (An-An, Kang et al. 2012). However, SIFT features relying on Gaussian filters are isotropic, meaning that they can struggle to characterize highly anisotropic adherent cells like myoblasts. The UWT-MIA method has the advantage over the SIFT method of being a multiscale transform-based texture analysis approach that can process at the same time information relative to the scale and the orientation of the cells (since it uses the Gabor continuous wavelet), producing interesting features to characterize anisotropic and irregular adherent cells such as myoblasts that tend to agglomerate at high confluence. Finally, cellular morphological features computed with the UWT-MIA (proposed method) are meaningful and have a physical sense (e.g. axis size in pixels or µm, orientation in degrees, roundness ratio), which makes data interpretation and analysis easier compared to raw SIFT features.

6.7 Conclusion

The UWT-MIA method is a pattern recognition image processing technique that can generate textural features that are related with cellular morphology. Using a set of 200 ground truth calibration images, PLS models were calibrated in order to estimate cellular morphological features distributions (major and minor axes, orientation and roundness distributions) from UWT-MIA textural features. The PLS models were then applied on a large set of PCM images time series (631,136 images acquired in 138 wells) to track the evolution of cellular morphological features throught time. A PCA model was also used to discrimate the morphology of myoblasts grown in two different media (serum supplemented or SSM, and serum free or SFM). It was found that cells grown in SFM were statistically smaller and more elongated than in SSM, which was detected by the proposed imaging technique.

The experimental results showed that the proposed method can be used on large datasets of PCM live-cell images captured under different conditions to assess cellular morphology and growth kinetics as a function of culture conditions and cell type. This technique could

thus be used for high-throughput screening applications (e.g. for the development of new culture media) and for cell culture monitoring and control in a non-invasive manner.

6.8 Acknowledgements

The authors would like to thank Prof. J. Jay Liu for providing codes and insights for computing 2D continuous wavelets. The authors would like also to acknowledge the lab of Dr. J.P. Tremblay (CRCHUL) for providing expertise/material for the culture of myoblasts. The work performed was partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT). The authors would like to acknowledge the Colosse team of Université Laval (managed by Calcul Québec and Compute Canada) for their help in processing the PCM images. The operation of the Colosse computer is funded by the Canada Foundation for Innovation (CFI), NanoQuébec, RMGA and the FQRNT.

Chapter 7. Conclusion and recommendations

7.1 Research problem

The development of new serum-free media for growing human cells is a key area of cell therapy research. Traditionnally fetal bovine serum has been used successfully as a key supplement to cultivate animal cells *in vitro*, since it contains proteins that promote cell proliferation and adhesion to substrates (Freshney 2010). Fetal bovine serum (FBS) is an animal blood extract which is a complex mixture of proteins and inorganic compounds that is not chemically defined. As a consequence, using FBS to produce human cells for cellular therapy involves issues, as its animal origin makes it prone to contamination (e.g. prions) and its composition can vary from batch to batch. These issues explain why developing serum-free media is important for human cell therapies.

However developping a new serum-free medium is difficult, as removing FBS from a given basal medium can select a sublineage not typical of a whole cell population, it can decrease or stop cellular growth and support fewer passages for finite cell lines (Freshney 2010). To improve the basal medium and get a performance similar to the FBS-supplemented medium, a tedious optimization process is often necessary to screen and test different supplements. To develop new culture media capable of expanding human cells without FBS, high throughput screening (HTS) is often performed to test a huge variety of molecules combinations (growth factors, cytokines, etc.) at different concentrations using design of experiment (DOE) methods in order to assess the impact of culture conditions on both cellular growth and cellular morphology. Because of the huge quantity of data and images that can be generated by each HTS experiment, multivariate data analysis techniques are required along robust and fast image processing algorithms.

From an image processing perspective, fluorescence microscopy used in microscopy-based HTS is easier to implement for single-cell segmentation since it provides images with a higher signal to noise ratio compared to brightfield or phase contrast microscopy (Ali, Gooding et al. 2007). Segmenting cells stained with a fluorophore can be achieved rapidly with high accuracy by conventional image processing techniques such as thresholding and the watershed transform. However, fluorescence microscopy used for HTS has several drawbacks, such as adding antibodies, dyes or fluorophore probes into a cell culture, which is an intrusive process that can affect cellular biology due to the

cytotoxicity of both the additives and the UV light. Also, because of the residual labeling molecules, the cells produced afterwards cannot be transplanted into patients. As a consequence, there is still a need to develop non-intrusive HTS methods relying on microscopy.

Phase-contrast microscopy (PCM) is a well-established tool to image living cells in biomaterials engineering research (Theriault, Walker et al. 2012), and has the advantage of being non-invasive to study cells that are sensitive to phototoxicity (Xiong, Chia et al. 2011). However segmenting individual cells in PCM images is difficult and prone to computational errors, because of the bright and uneven halo surrounding them, blurry cell-cell interfaces (at a high degree of confluence), broken borders and poor intensity contrast between the background and cellular regions (Bradbury and Wan 2010). Several methods were proposed to achieve single-cell segmentation in PCM (e.g. level sets, nested kernels), but these methods are difficult to use widely for HTS because of their complexity, high computation times and their sensitivity to image quality (e.g. uneven background). Also, several methods developed for round cells (such as convolution template matching) do not work well for highly anisotropic adherent cells. Therefore, the aim of this PhD project was to develop robust and reliable image processing algorithms for PCM-based high-throughput screening applications (non-intrusive method).

As seen in the literature review (Chapter 2), several methods are available for analyzing microscopic images and choosing one or the other depends upon image quality, the microscopy technique used and even the type of cells used (adherent versus suspended cells). In HTS applications, critical information relies on cellular counts and cellular morphology, and image processing algorithms are required to analyze the large quantities of data generated. There are basically two approaches to detect cells and characterize their morphology: single-cell segmentation (using for instance edge or region detection) or pattern recognition relying on image features. In a single-cell segmentation approach, a segmentation map needs to be generated by performing a classification (via a classifier, an edge detection or a region detection method) using the raw image or local spatial features. Once a segmentation map is generated, it is possible to locate precisely each cell, to compute a cell count and to compute morphological features. A second approach is the pattern recognition approach, in which cell counts and cell morphological features are inferred directly from the raw image or from features via a classifier or using regression

and/or multivariate methods (e.g. PLS). The main difference is that pattern recognition approaches do not require a segmentation map, thus reducing the influence of noise on the results while being more computation efficient (compared to level sets for example). However the disadvantage of pattern recognition approaches is that accurate cell localization is not always possible, and thus cell tracking is usually not possible.

Since single-cell segmentation is difficult to achieve when imaging adherent cells in PCM-related high throughput screening applications, pattern recognition approaches were used in this PhD project to assess the cell-covered surface, cellular count and cellular morphology of myoblasts relative to the culture medium used. The main goal of this PhD project was to be able to assess cellular growth kinetics and cellular morphology relative to culture conditions in PCM-based high throughput screening experiments. The main problems to solve were to develop image processing stategies that are robust and fast so they can be used on a large scale, as an important quantity of images are generated by each HTS experiment.

7.2 Contributions

The first issue to tackle was to assess the growth kinetics of adherent cells in PCM images in a quick and robust manner (Chapter 4). To avoid single-cell segmentation, the evolution of the cell-covered surface through time was used to quantify cellular growth, as the surface covered by the cells (the degree of confluence) is roughly proportional to the number of cells. The growth kinetics of adherent cells in phase contrast microscopy was characterized in Chapter 4 using the cell-covered surface estimated using image features computed by the proposed range filter algorithm (based on the local range of values in the images). Since areas covered by cells have a texture incorporating higher local frequencies than background regions, a range filter properly tuned (combined with a size filter) was capable of detecting cell-covered areas in each image (using this distinctive pattern signature). Analyzing the evolution of the cell-covered surface through time, we showed that myoblasts grown in SSM and myoblasts grown in SFM have similar growth kinetics, and we established a statistical link between the growth kinetics parameters and the cells passage number. The cell-covered surface approach was already used in the literature, but we generalized it in Chapter 4 to take into account the adjustment of the parameters of different algorithms proposed by several authors to estimate the cell-covered surface (commonly referred to as the degree of confluence). Even if there is a growing number of reports about image processing algorithms for PCM images, it is

difficult to compare them directly because each algorithm is designed for a specific problem, providing a solution for a specific imaging modality, microorganism or cell type, and tuned on a particular benchmark dataset. The main contribution of Chapter 4 relies on the development of a method based on a pattern recognition metric (the F-measure function) to tune the parameters of each algorithm available in order to compare them (using a set of 200 ground truth images). Several algorithms provided in the literature were tested along the proposed range filter method to compute the cell-covered surface of 200 ground truth images, and most of the proposed algorithms provided roughly a similar performance when properly tuned with the F-measure function. However, the results showed that the range filter algorithm provided a slightly better performance in terms of R and P than the other algorithms while having a competitive average computing time per image (ACT), thus justifying its selection.

The second research challenge was to extract local pixel level textural patterns within PCM images that are relevant to cellular morphology (Chapter 5), and to quantify the cells orientations and sizes. Classical multivariate image analysis (MIA) methods can group pixels in RGB or multispectral images together according to their respective spectral patterns (e.g. color) via principal component analysis (PCA). However, PCM images acquired by conventional microscopes are often monochrome (grey scale image, as it is the case in this project), and there is no spectral information available. Also, conventional MIA methods cannot be used to analyze textural information within images without losing spatial information due to the PCA unfolding operation. To analyze local textural patterns in image datasets from a multivariate perspective, two approaches were developed: shifting and stacking (Bharati, Liu et al. 2004) and MR-MIA methods (Liu and MacGregor 2007). Shifting and stacking can be prone to noise because it uses raw pixel values and is computationally intensive since it may require a lot of shifted version of the image to capture with enough resolution all objects scales and orientations. Since MR-MIA methods use the decimated wavelet transform (DWT), this approach requires to build several multivariate models (PCAs) to explore each scale because of the lack of congruency between detail coefficients matrices. As a consequence, there was a need to develop a new method that could address these limitations. In Chapter 5, a new MIA method relying on the undecimated wavelet transform (UWT-MIA) was proposed. The continuous Gabor wavelet transform was used, providing an optimal resolution to analyze the texture within PCM images from a scale and orientation perspective, while considering spectral

information if needed. The main contribution of Chapter 5 consists of the use of a continuous undecimated wavelet transform coupled to conventional MIA analysis techniques (e.g. score density histograms) to detect both spectral and textural patterns within image datasets at the pixel level. Stacking congruent wavelet planes from the undecimated wavelet transform represents an improvement compared to the MR-MIA methods, as a single cube of data is generated to analyze all objects scales and orientations at the same time in a single PCA model. This provides the capacity to summarize for instance the information related to adherent cells orientations and sizes in a single score density histogram. As shown in Chapter 5, the UWT-MIA method could be used beyond PCM-related applications. For instance, it was used to analyze different sets of RGB images to highlight both color and textual patterns. As a result, the UWT-MIA method represents a significant contribution to the field of multivariate image analysis in general.

The third contribution of this thesis was to establish a relationship between the local textural patterns extracted within PCM images using the new UWT-MIA and cellular morphological features (Chapter 6). The main challenge was to find a way to establish a statistical link between pixel counts in score density histograms and cellular morphological features. Pixel counts under defined masks in the $\mathbf{T_1/T_2}$ score density histograms generated by the UWT-MIA method were used (**X** matrix). A set of 200 ground truth images was used to generate distributions of morphological features (major and minor axes, orientation and roundness) that were stored in a response matrix **Y**. PLS models were built to predict morphological features distributions from pixel counts under masks in $\mathbf{T_1/T_2}$, maximizing both R^2_{CAL} and R^2_{VAL} by grid search while adjusting the mask parameters and the number of PLS components. The main contribution of Chapter 6 was to develop a method relying on the UWT-MIA to compute morphological features on a large dataset of PCM images to relate the shape of the cells to culture conditions. In Chapter 6, the new method developed (UWT-MIA) was used to analyze the image database (around 1 million PCM images of myoblasts). The morphological features computed (major and minor axes, roundness and orientation distributions) were used to highlight the influence of the culture medium on the morphology of myoblasts. A PCA analysis showed that cells grown in SFM were smaller and more elongated than cells grown in SSM, and these results were found consistent with qualitative observations. The pattern recognition method proposed in Chapter 6, which is both computationally efficient and precise, can be used to process

PCM images in HTS and cell culture control applications (since it does not require single-cell segmentation). The next step would be to use these image processing techniques to establish a direct link between cellular morphology and cellular type and functionality (related for instance to the expression of surface markers, as described in Section 2.2.3). This problem that was not addressed directly in this PhD project.

One of the main advantages of using multivariate techniques (e.g. PCA, PLS) to analyze morphological features computed from the UWT-MIA method is their interpretability compared to other classifiers such as neural networks and Markov Random Fields. It is easy to interpret the importance of the variables in the model by looking at the VIPs or at the loadings plot (see Figure 6-12). Also, the morphological features estimated from the UWT-MIA features have a physical meaning (e.g. average major axis in µm, mean orientation in degrees, etc.), which makes data easier to interpret by experimental cell biologists.

7.3 Perspectives

Image processing methods developed for this PhD project (in Chapter 4, Chapter 5 and Chapter 6) were able to highlight patterns in large image datasets, making it possible to assess the impact of culture conditions on cellular growth kinetics and morphologies in an automated fashion. These methods could be used on other cellular phenotypes in high-throughput screening and cell culture control applications. Among cell culture control applications, tissue engineering and cell therapy could benefit from these image processing techniques, providing useful information in terms of cell quality (related to textural analysis and cell morphology) and cell quantity (assessment of cell counts) while being non-intrusive (relying on PCM). Among cell therapy research fields that could benefit from these image processing methods, it is possible to consider the production of keratinocytes for skin grafts, the culture of healthy myoblasts to treat dystrophic patients and the expansion of endothelial cells to repair vascular tissues.

From a cell production perspective, the image processing techniques developed in this PhD project can be used for online cell culture control when the cells are grown on flat surfaces (e.g. multiwell plates, Petri dishes or T-flasks) and are imaged in PCM. However the possibility of developing a microscopy set-up to acquire PCM images of cells growing on the surface of rolling bottles should be investigated in future projects, as they constitute a mean of producing larger quantities of cells. Another way of producing larger quantities

of cells is to grow them on suspended microbeads, but acquiring PCM images in that case would be difficult or even impossible.

Even though the image processing methods developed in this PhD project open the door to new applications, there are still several issues and problems that should be addressed in future work. For instance, actual computation times required to calculate morphological features via the UWT-MIA method (as in Chapter 6) are around 30 seconds/image, and processing 121 images (one well scanned with a grid of 11x11) could take up to 1 hour on a standard computer. This limitation is not related to the method itself, but more to memory limitations and to the important number of wavelet planes generated (400 per image) that need to be stacked and analyzed. There was no tentative to decrease the number of wavelet planes during this PhD project as the main focus was on the development of the method and not on minimizing computation time. Since the main bottlenecks are the number of wavelet planes used, the number of masks used and also memory limitations, future work could focus on computational efficiency. Because of memory limitations (RAM), it is not possible to load a complete wavelet planes cube (WPC) having a significant number of wavelet planes: it must be loaded into Matlab® in several steps, one fraction at a time, increasing computation times significantly. One improvement that should be investigated is the use of variable selection methods to only compute wavelet planes that are the most critical to predict morphological features. To increase computation speed, the algorithms could be implemented in C++. Parallelization could also be considered, as it was used to speed-up the computations required to generate results in Chapter 6. For instance, the super computer Colosse has several nodes (8 processors each) that can be used for processing large batches of images (up to 40 images at the same time, by experience). Even with these memory limitations, the proposed method (in its current state, without wavelet planes selection) can still be used for HTS applications, since large batches of images can be processed offline (after the experiment is done, accepting a lag in the results) or even online, accepting a processing time of 30 seconds/image. Also, for the purpose of this project a grid of 11x11 images was acquired for each time acquisition, but sub-sampling could be considered to decrease the overall computation times while providing morphological features estimates that are still representative of the whole cell population.

From a tuning perspective, additionnal topics such as inter-observer variability and the choice of the optimization method could be addressed in future work. In Chapter 4 and in

Chapter 6, only one ground truth dataset of 200 images (built by one observer) was used, and it would be interesting to assess the inter-observer variability by asking other scientists to perform a segmentation on the same set of PCM pictures. To adjust the parameters associated with the proposed image processing algorithms, grid search was used to choose the optimal settings over a specified range of values. Other machine learning methods (such as evolutionary algorithms) could be tested to see if they could provide an optimal set of parameters in a more flexible manner. To explore in more details the scores space generated by the UWT-MIA and to build masks that would be even more precise (in more than 2 dimensions for instance), machine learning (such as SVM, as described in Liu et al. (2005)) could be useful.

Future projects could also explore the potential of combining conventional segmentation methods and the new UWT-MIA method (described in Chapter 5). For instance, the UWT-MIA approach could be used to produce background and object markers that could be fed to a watershed transform algorithm (see Section 2.4.4.2.4) in order to perform single-cell segmentation if the later is desired. With the actual experimental set-up, stacks of brightfield and PCM images at different focal planes around the specimen could be acquired in order to do quantitative phase microscopy. PCA and PLS methods could be used on these cubes of PCM or brightfield images to extract valuable information, and classify pixels between background and cellular regions. In future projects, fluorophores could be used in order to combine the information related to the presence or absence of markers (by processing images acquired in epifluorescence) with the information related to cell morphology (by processing PCM images). Both information could be colocalized and could be used to establish a statistical link between cellular phenotype and cell morphology.

References

Acton, S. T., C. Yang, et al. (2009). Poisson inverse gradient approach to vascular myocyte detection and segmentation. Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on.

Adams, R. and L. Bischof (1994). "Seeded region growing." Pattern Analysis and Machine Intelligence, IEEE Transactions on **16**(6): 641-647.

Agley, C. C., C. P. Velloso, et al. (2012). "An Image Analysis Method for the Precise Selection and Quantitation of Fluorescently Labeled Cellular Constituents: Application to the Measurement of Human Muscle Cells in Culture." Journal of Histochemistry & Cytochemistry **60**(6): 428-438.

Akaike, H. (1974). "A new look at the statistical model identification." Automatic Control, IEEE Transactions on **19**(6): 716-723.

Al-Muhairi, H., M. Fleury, et al. (2011). Parameter-orientated segmentation algorithm evaluation. GCC Conference and Exhibition (GCC), 2011 IEEE.

Al-Shanti, N., S. H. Faulkner, et al. (2011). "A Semi-automated Programme for Tracking Myoblast Migration Following Mechanical Damage: Manipulation by Chemical Inhibitors." Cellular Physiology and Biochemistry **27**(6): 625-636.

Ali, R., M. Gooding, et al. (2007). PHASE-BASED SEGMENTATION OF CELLS FROM BRIGHTFIELD MICROSCOPY. Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on.

Ambriz-Colin, F., M. Torres-Cisneros, et al. (2006). Detection of Biological Cells in Phase-Contrast Microscopy Images. Artificial Intelligence, 2006. MICAI '06. Fifth Mexican International Conference on.

Ambuhl, M. E., C. Brepsant, et al. (2012). "High-resolution cell outline segmentation and tracking from phase-contrast microscopy images." Journal of Microscopy **245**(2): 161-170.

An-An, L., L. Kang, et al. (2012). "A Semi-Markov Model for Mitosis Segmentation in Time-Lapse Phase Contrast Microscopy Image Sequences of Stem Cell Populations." Medical Imaging, IEEE Transactions on **31**(2): 359-369.

Anan, L., L. Kang, et al. (2010). Spatiotemporal mitosis event detection in time-lapse phase contrast microscopy image sequences. Multimedia and Expo (ICME), 2010 IEEE International Conference on.

Andrle, R. (1994). "THE ANGLE MEASURE TECHNIQUE - A NEW METHOD FOR CHARACTERIZING THE COMPLEXITY OF GEOMORPHIC LINES." Mathematical Geology **26**(1): 83-97.

Athelogou, M., M. Eblenkamp, et al. (2010). Single-cell detection and classification in phase contrast images. Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on.

Basu, S. and S. T. Acton (2009). Object segmentation by traversing a pose-shape manifold. Image Processing (ICIP), 2009 16th IEEE International Conference on.

Bentzinger, C. F., Y. X. Wang, et al. (2012). "Building Muscle: Molecular Regulation of Myogenesis." Cold Spring Harbor Perspectives in Biology **4**(2).

Berlage, T. (2005). "Analyzing and mining image databases." Drug Discovery Today **10**(11): 795-802.

Berlage, T. (2007). "Analyzing and mining automated imaging experiments." Expert Opinion on Drug Discovery **2**(4): 561-569.

Bharati, M. H., J. J. Liu, et al. (2004). "Image texture analysis: methods and comparisons." Chemometrics and Intelligent Laboratory Systems **72**(1): 57-71.

Boudreault, P., J. P. Tremblay, et al. (2001). "Scale-up of a myoblast culture process." Journal of Biotechnology **91**(1): 63-74.

Bradbury, L. and J. W. L. Wan (2010). A spectral k-means approach to bright-field cell image segmentation. Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE.

Bradhurst, C. J., W. Boles, et al. (2008). Segmentation of bone marrow stromal cells in phase contrast microscopy images. Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference.

Brenner, S. (2008). "An interview with... Sydney Brenner. Interview by Errol C. Friedberg." Nature reviews. Molecular cell biology **9**(1): 8-9.

Brinkmann, M., D. Lutkemeyer, et al. (2002). "New technologies for automated cell counting based on optical image analysis 'The Cellscreen'." Cytotechnology **38**(1-2): 119-127.

Buckingham, M., L. Bajard, et al. (2003). "The formation of skeletal muscle: from somite to limb." Journal of Anatomy **202**(1): 59-68.

Burgemeister, S., T. W. Nattkemper, et al. (2010). "CellViCAM-Cell viability classification for animal cell cultures using dark field micrographs." Journal of Biotechnology **149**(4): 310-316.

ByoungChul, K., S. MiSuk, et al. (2007). Microscopic Image Segmentation for the Clinical Support System. Computer Graphics, Imaging and Visualisation, 2007. CGIV '07.

Caldwell, M. A., E. Garcion, et al. (2004). "Heparin stabilizes FGF-2 and modulates striatal precursor cell behavior in response to EGF." Experimental Neurology **188**(2): 408-420.

Canny, J. (1986). "A Computational Approach to Edge Detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI-8**(6): 679-698.

Cantarella, C., L. Sepe, et al. (2009). "Analysis and modelling of motility of cell populations with MotoCell." Bmc Bioinformatics **10**.

Caponetti, L., G. Castellano, et al. (2009). Multiresolution texture analysis for human oocyte cytoplasm description. Medical Measurements and Applications, 2009. MeMeA 2009. IEEE International Workshop on.

Carpenter, A. E., T. R. Jones, et al. (2006). "CellProfiler: image analysis software for identifying and quantifying cell phenotypes." Genome Biology **7**(10).

Chan, T. F. and L. A. Vese (2001). "Active contours without edges." Image Processing, IEEE Transactions on **10**(2): 266-277.

Chance, B., B. Schoener, et al. (1979). "Oxidation-reduction ratio studies of mitochondria in freeze-trapped samples - NADH and flavoprotein fluorescence signals." Journal of Biological Chemistry **254**(11): 4764-4771.

Chao-Hui, H., S. Sankaran, et al. (2012). "Online 3-D Tracking of Suspension Living Cells Imaged with Phase-Contrast Microscopy." Biomedical Engineering, IEEE Transactions on **59**(7): 1924-1933.

Charge, S. B. P. and M. A. Rudnicki (2004). "Cellular and molecular regulation of muscle regeneration." Physiological Reviews **84**(1): 209-238.

Cheng, J. and J. C. Rajapakse (2009). "Segmentation of Clustered Nuclei With Shape Markers and Marking Function." Ieee Transactions on Biomedical Engineering **56**(3): 741-748.

Chowdhury, M. H. and W. D. Little (1995). Image thresholding techniques. Communications, Computers, and Signal Processing, 1995. Proceedings. IEEE Pacific Rim Conference on.

Chunni, D., L. Qingli, et al. (2010). Evaluation of Erythropoietin Efficacy on Diabetic Rats Based on Microscopic Hyperspectral Imaging Technique. Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on.

Constantinou, P., R. S. Dacosta, et al. (2009). "Extending immunofluorescence detection limits in whole paraffin-embedded formalin fixed tissues using hyperspectral confocal fluorescence imaging." Journal of Microscopy-Oxford **234**(2): 137-146.

Cox, G. (2007). Optical imaging techniques in cell biology. Boca Raton, CRC/Taylor & Francis.

Curl, C. L., T. Harris, et al. (2004). "Quantitative phase microscopy: a new tool for measurement of cell culture growth and confluency in situ." Pflugers Archiv-European Journal of Physiology **448**(4): 462-468.

Cutler, P. J., D. M. Haaland, et al. (2009). "Systematic Method for the Kinetic Modeling of Temporally Resolved Hyperspectral Microscope Images of Fluorescently Labeled Cells." Applied Spectroscopy **63**(3): 261-270.

Danoviz, M. and Z. Yablonka-Reuveni (2012). Skeletal Muscle Satellite Cells: Background and Methods for Isolation and Analysis in a Primary Culture System. Myogenesis. J. X. DiMario, Humana Press. **798:** 21-52.

Dashan, G., D. Padfield, et al. (2010). Microscopy image focus quality assessment by overlapping tile analysis. Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on.

Davis, J. and M. Goadrich (2006). The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine learning. Pittsburgh, Pennsylvania, ACM**:** 233-240.

Davis, P. J., E. A. Kosmacek, et al. (2007). "The Large-Scale Digital Cell Analysis System: an open system for nonperturbing live cell imaging." Journal of Microscopy **228**(3): 296-308.

Deasy, B. M., R. J. Jankowski, et al. (2001). "Muscle-derived stem cells: Characterization and potential for cell-mediated therapy." Blood Cells Molecules and Diseases **27**(5): 924-933.

Deasy, B. M., R. J. Jankowski, et al. (2003). "Modeling stem cell population growth: Incorporating terms for proliferative heterogeneity." Stem Cells **21**(5): 536-545.

Debeir, O., I. Adanja, et al. (2008). Phase contrast image segmentation by weak watershed transform assembly. Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on.

Debeir, O., V. Megalizzi, et al. (2008). "Videomicroscopic extraction of specific information on cell proliferation and migration in vitro." Experimental Cell Research **314**(16): 2985-2998.

Debeir, O., P. Van Ham, et al. (2005). "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes." Medical Imaging, IEEE Transactions on **24**(6): 697-711.

Dejong, S. (1993). "SIMPLS - AN ALTERNATIVE APPROACH TO PARTIAL LEAST-SQUARES REGRESSION." Chemometrics and Intelligent Laboratory Systems **18**(3): 251-263.

Demirkaya, O., M. H. Asyali, et al. (2009). Image processing with MATLAB : applications in medicine and biology. Boca Raton, CRC Press/Taylor & Francis.

Demos, S. G., C. A. Lieber, et al. (2005). "Imaging of tissue microstructures using a multimodal microscope design." Selected Topics in Quantum Electronics, IEEE Journal of **11**(4): 752-758.

DeVaney, T. (2008). "Plate Imaging Inserts for Meniscus Correction."

Dewan, M. A. A., M. O. Ahmad, et al. (2011). "Tracking Biological Cells in Time-Lapse Microscopy: An Adaptive Technique Combining Motion and Topological Features." Biomedical Engineering, IEEE Transactions on **58**(6): 1637-1647.

Duchesne, C., J. J. Liu, et al. (2012). "Multivariate image analysis in the process industries: A review." Chemometrics and Intelligent Laboratory Systems **117**: 116-128.

Duchesne, C. and J. F. MacGregor (2001). "Jackknife and bootstrap methods in the identification of dynamic models." Journal of Process Control **11**(5): 553-564.

Dunn, K. G. (2010). MACC Multivariate Image Analysis (MACCMIA). Hamilton, Ontario, Canada, McMaster University.

Eilken, H. M., S. I. Nishikawa, et al. (2009). "Continuous single-cell imaging of blood generation from haemogenic endothelium." Nature **457**(7231): 896-900.

Eliceiri, K. W., M. R. Berthold, et al. (2012). "Biological imaging software tools." Nature Methods **9**(7): 697-710.

Eriksson, L., N. Kettaneh-Wold, et al. (2001). Multi- and megavariate data analysis - principles and applications. Umea, Umetrics Academy.

Ersoy, I., F. Bunyak, et al. (2008). Cell segmentation using Hessian-based detection and contour evolution with directional derivatives. Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.

Facco, P., A. Masiero, et al. (2011). "Improved multivariate image analysis for product quality monitoring." Chemometrics and Intelligent Laboratory Systems **109**(1): 42-50.

Failmezger, H., H. Frohlich, et al. (2013). "Unsupervised automated high throughput phenotyping of RNAi time-lapse movies." Bmc Bioinformatics **14**.

Fatima, M. M. and V. Seenivasagam (2011). A Fast Fuzzy-C means based marker controlled watershed segmentation of clustered nuclei. Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on.

Fenster, A. and B. Chiu (2005). Evaluation of Segmentation algorithms for Medical Imaging. Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.

Freshney, I. R. (2000). Culture of animal cells : a manual of basic technique New York, Wiley-Liss.

Freshney, R. I. (2010). Culture of Animal Cells: A Manual of Basic Technique and Specialized Applications, John Wiley & Sons.

Frigault, M. M., J. Lacoste, et al. (2009). "Live-cell microscopy - tips and tools." Journal of Cell Science **122**(6): 753-767.

Gao, L. A., R. T. Kester, et al. (2010). "Snapshot Image Mapping Spectrometer (IMS) with high sampling density for hyperspectral microscopy." Optics Express **18**(14): 14330-14344.

Geladi, P. (1992). "Some special topics in multivariate image analysis." Chemometrics and Intelligent Laboratory Systems **14**(1-3): 375-390.

Geladi, P. and H. Grahn (1996). Multivariate Image Analysis. New-York, John Wiley & Sons Ltd.

Geladi, P. and B. R. Kowalski (1986). "PARTIAL LEAST-SQUARES REGRESSION - A TUTORIAL." Analytica Chimica Acta **185**: 1-17.

Genovesio, A., T. Liedl, et al. (2006). "Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots." Ieee Transactions on Image Processing **15**(5): 1062-1070.

Gerardi, S., G. Galeazzi, et al. (2005). "A microcollimated ion beam facility for investigations of the effects of low-dose radiation." Radiation Research **164**(4 Pt 2): 586-590.

Golay, M. J. E. and A. Savitzky (1964). "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." Analytical chemistry **36**(8): 1627-1639.

Goldman, R. D. and D. L. Spector (2005). Live Cell Imaging: A Laboratory Manual, Cold Spring Harbor Laboratory Press.

Gonzalez, R. C. and R. E. Woods (2008). Digital image processing. Upper Saddle River, NJ, Pearson/Prentice Hall.

Gosselin, R. (2010). On-line quality control in polymer processing using hyperspectral imaging. Génie Chimique. Québec, Université Laval. **PhD:** 223 p.

Gosselin, R., D. Rodrigue, et al. (2010). "A Bootstrap-VIP approach for selecting wavelength intervals in spectral imaging applications." Chemometrics and Intelligent Laboratory Systems **100**(1): 12-21.

Haaland, D. M., H. D. T. Jones, et al. (2009). "Hyperspectral Confocal Fluorescence Imaging: Exploring Alternative Multivariate Curve Resolution Approaches." Applied Spectroscopy **63**(3): 271-279.

Ham, R. G. and H. M. Blau (1988). "Improved Media for Normal Human Muscle Satellite Cells: Serum-Free Clonal Growth and Enhanced Growth with Low Serum." In Vitro Cellular & Developmental Biology **24**(8): 833-844.

Hamahashi, S., S. Onami, et al. (2005). "Detection of nuclei in 4D Nomarski DIC microscope images of early Caenorhabditis elegans embryos using local image entropy and object tracking." Bmc Bioinformatics **6**.

Hand, A. J., T. Sun, et al. (2009). "Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration." Journal of Microscopy-Oxford **234**(1): 62-79.

Hand, D. J. (2012). "Assessing the Performance of Classification Methods." International Statistical Review **80**(3): 400-414.

Haralick, R. M., R. Haralick, et al. (1973). "Textural Features for Image Classification." IEEE Transactions on Systems, Man and Cybernetics **SMC3**(6): 610-621.

Haralick, R. M., S. R. Sternberg, et al. (1987). "Image Analysis Using Mathematical Morphology." Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI-9**(4): 532-550.

He, W. J., X. X. Wang, et al. (2007). Cell segmentation for division rate estimation in computerized video time-lapse microscopy. Multimodal Biomedical Imaging Ii. F. S. Azar. **6431**.

Heid, P. J., E. Voss, et al. (2002). "3D-DIASemb: A computer-assisted system for reconstructing and motion analyzing in 4D every cell and nucleus in a developing embryo." Developmental Biology **245**(2): 329-347.

Held, M., M. H. A. Schmitz, et al. (2010). "CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging." Nature Methods **7**(9): 747-754.

Hoesli, C. A., A. Garnier, et al. (2014). "A fluorophore-tagged RGD peptide to control endothelial cell adhesion to micropatterned surfaces." Biomaterials **35**(3): 879-890.

Hoffman, E., R. Brown, et al. (1987). "Dystrophin: The protein product of the duchenne muscular dystrophy locus." Cell **51**(6): 919-928.

Horowitz, S. L. and T. Pavlidis (1976). "PICTURE SEGMENTATION BY A TREE TRAVERSAL ALGORITHM." Journal of the Acm **23**(2): 368-388.

Hou, Z., S. Ma, et al. (2007). Segmentation and Recognition of Marrow Cells Image Based on Wavelet and Genetic Cluster. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on.

House, D., M. L. Walker, et al. (2009). Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli. Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on.

Hui, X., R. Weibin, et al. (2006). Wavelet-Based Focus Measure and 3-D Surface Reconstruction Method for Microscopy Images. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.

Huiming, P., Z. Xiaobo, et al. (2009). Integrating multi-scale blob/curvilinear detector techniques and multi-level sets for automated segmentation of stem cell images. Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on.

Huth, J., M. Buchholz, et al. (2011). "TimeLapseAnalyzer: Multi-target analysis for live-cell imaging and time-lapse microscopy." Computer Methods and Programs in Biomedicine **104**(2): 227-234.

Huth, J., M. Buchholz, et al. (2010). "Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system." Bmc Cell Biology **11**.

Jackson, D. (1993). "Stopping Rules in Principal Components Analysis: A Comparison of Heuristical and Statistical Approaches." Ecology **74**(8): 2204-2214.

Jacques, L., A. Coron, et al. (2007). "YAWTb : Yet Another Wavelet Toolbox." 2011, from http://sites.uclouvain.be/ispgroup/yawtb/pmwiki.php/Main/Presentation.

Jain, A. K., R. P. W. Duin, et al. (2000). "Statistical pattern recognition: A review." Ieee Transactions on Pattern Analysis and Machine Intelligence **22**(1): 4-37.

Jain, A. K. and F. Farrokhnia (1991). "UNSUPERVISED TEXTURE SEGMENTATION USING GABOR FILTERS." Pattern Recognition **24**(12): 1167-1186.

Jiang, R. M., D. Crookes, et al. (2010). "Live-Cell Tracking Using SIFT Features in DIC Microscopic Videos." Biomedical Engineering, IEEE Transactions on **57**(9): 2219-2228.

Jun, K., L. Cooper, et al. (2010). Texture based image recognition in microscopy images of diffuse gliomas with multi-class gentle boosting mechanism. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.

Juneau, P.-M., A. Garnier, et al. (2015). "The undecimated wavelet transform–multivariate image analysis (UWT-MIA) for simultaneous extraction of spectral and spatial information." Chemometrics and Intelligent Laboratory Systems **142**(0): 304-318.

Juneau, P. M., A. Garnier, et al. (2013). "Selection and Tuning of a Fast and Simple Phase-Contrast Microscopy Image Segmentation Algorithm for Measuring Myoblast Growth Kinetics in an Automated Manner." Microscopy and Microanalysis **19**(4): 855-866.

Kachouie, N. N., P. Fieguth, et al. (2010). "Constrained Watershed Method to Infer Morphology of Mammalian Cells in Microscopic Images." Cytometry Part A **77A**(12): 1148-1159.

Kachouie, N. N., P. Fieguth, et al. (2007). Stem-Cell Localization: A Deconvolution Problem. Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE.

Kachouie, N. N., P. Fieguth, et al. (2008). Watershed deconvolution for cell segmentation. Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE.

Kachouie, N. N., L. J. Lee, et al. (2005). A probabilistic living cell segmentation model. Image Processing, 2005. ICIP 2005. IEEE International Conference on.

Kang, L., E. D. Miller, et al. (2006). Online Tracking of Migrating and Proliferating Cells Imaged with Phase-Contrast Microscopy. Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on.

Kapur, J. N., P. K. Sahoo, et al. (1985). "A new method for gray-level picture thresholding using the entropy of the histogram." Computer Vision, Graphics, and Image Processing **29**(3): 273-285.

Kass, M., A. Witkin, et al. (1988). "Snakes: Active contour models." International Journal of Computer Vision **1**(4): 321-331.

Katari, S., M. Wallack, et al. (2009). "Fabrication and evaluation of a near-infrared hyperspectral imaging system." Journal of Microscopy-Oxford **236**(1): 11-17.

Kenong, W., D. Gauthier, et al. (1995). "Live cell image segmentation." Biomedical Engineering, IEEE Transactions on **42**(1): 1-12.

Ker, D. F. E., L. E. Weiss, et al. (2011). "An Engineered Approach to Stem Cell Culture: Automating the Decision Process for Real-Time Adaptive Subculture of Stem Cells." Plos One **6**(11).

Kingdom, F. and A. Olmos McGill Calibrated Colour Image Database, McGill Vision Research.

Kittler, J. and J. Illingworth (1986). "Minimum error thresholding." Pattern Recognition **19**(1): 41-47.

Kreyszig, E. (2000). Advanced Engineering Mathematics, Maple Computer Guide, John Wiley & Sons.

Lankton, S. M. (2009). LOCALIZED STATISTICAL MODELS IN COMPUTER VISION, Georgia Institute of Technology. **PhD**.

Leysi-Derilou, Y. (2011). MONITORING AND MATHEMATICAL MODELING OF IN VITRO HUMAN MEGAKARYOCYTE EXPANSION AND MATURATION DYNAMICS. Chemical Engineering. Québec, Laval. **PhD**.

Leysi-Derilou, Y., C. Duchesne, et al. (2012). "Single-cell level analysis of megakaryocyte growth and development." Differentiation **83**(4): 200-209.

Lied, T. T., P. Geladi, et al. (2000). "Multivariate image regression (MIR): implementation of image PLSR—first forays." Journal of Chemometrics **14**(5-6): 585-598.

Lim, J. H. F. and G. A. Davies (1990). "A STOCHASTIC-MODEL TO SIMULATE THE GROWTH OF ANCHORAGE DEPENDENT CELLS ON FLAT SURFACES." Biotechnology and Bioengineering **36**(6): 547-562.

Liow, Y. T. and T. Pavlidis (1988). Enhancements of the split-and-merge algorithm for image segmentation. Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on.

Liu, A., K. Li, et al. (2011). "A Semi-Markov Model for Mitosis Segmentation in Time-Lapse Phase Contrast Microscopy Image Sequences of Stem Cell Populations." Medical Imaging, IEEE Transactions on **PP**(99): 1-1.

Liu, J. (2004). Machine vision for process industries: monitoring, control, and optimization of visual quality of processes and products. School of Graduate Studies. Hamilton, Ontario, McMaster University. **PhD:** 170.

Liu, J., W. Gui, et al. (2011). Production conditions classification for froth flotation based on froth image processing. Control Conference (CCC), 2011 30th Chinese.

Liu, J. and J. MacGregor (2008). "Froth-based modeling and control of flotation processes." Minerals Engineering **21**(9): 642-651.

Liu, J. J., M. H. Bharati, et al. (2005). "Automatic masking in multivariate image analysis using support vector machines." Chemometrics and Intelligent Laboratory Systems **79**(1-2): 42-54.

Liu, J. J. and J. F. MacGregor (2007). "On the extraction of spectral and spatial information from images." Chemometrics and Intelligent Laboratory Systems **85**(1): 119-130.

Liu, J. P., W. H. Gui, et al. (2013). "Machine Vision Based Production Condition Classification and Recognition for Mineral Flotation Process Monitoring." International Journal of Computational Intelligence Systems **6**(5): 969-986.

Liu, T., J. Nie, et al. (2008). "ZFIQ: a software package for zebrafish biology." Bioinformatics **24**(3): 438-439.

Long, X., W. L. Cleveland, et al. (2005). "A new preprocessing approach for cell recognition." Information Technology in Biomedicine, IEEE Transactions on **9**(3): 407-412.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on.

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." International Journal of Computer Vision **60**(2): 91-110.

Lu, C., X. Bai, et al. (2008). An efficient image segmentation method with application to cell images. Signal Processing, 2008. ICSP 2008. 9th International Conference on.

Mallat, S. G. (1989). "A theory for multiresolution signal decomposition: the wavelet representation." Pattern Analysis and Machine Intelligence, IEEE Transactions on **11**(7): 674-693.

Malpica, N., A. Santos, et al. (2003). "Automatic quantification of viability in epithelial cell cultures by texture analysis." Journal of Microscopy-Oxford **209**: 34-40.

McConico, M. B., R. B. Horton, et al. (2012). "Monitoring chemical impacts on cell cultures by means of image analyses." Journal of Chemometrics **26**(11-12): 585-597.

Min, C., C. Shengyong, et al. (2009). Hybrid Contour Model for Segmentation of Cell Nucleolus and Membranes. Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on.

MIR. (1999). "A Glossary of Photographic Terms." Retrieved May 26th, 2014, from http://www.mir.com.my/rb/photography/glossary/terms_f.htm.

Miura, K. (2005). Tracking movement in cell biology. Microscopy Techniques. J. Rietdorf. Berlin, Springer-Verlag Berlin. **95:** 267-295.

Mõlder, A., M. Sebesta, et al. (2008). "Non-invasive, label-free cell counting and quantitative analysis of adherent cells using digital holography." Journal of Microscopy **232**(2): 240-247.

Monici, M. (2005). Cell and tissue autofluorescence research and diagnostic applications. Biotechnology Annual Review. M. R. El-Gewely, Elsevier. **Volume 11:** 227-256.

Montarras, D., J. Morgan, et al. (2005). "Direct isolation of satellite cells for skeletal muscle regeneration." Science **309**(5743): 2064-2067.

Moogk, D., M. Stewart, et al. (2010). "Human ESC Colony Formation Is Dependent on Interplay Between Self-Renewing hESCs and Unique Precursors Responsible for Niche Generation." Cytometry Part A **77A**(4): 321-327.

Mortensen, P. P. and K. H. Esbensen (2005). "Optimization of the Angle Measure Technique for image analytical sampling of particulate matter." Chemometrics and Intelligent Laboratory Systems **75**(2): 219-229.

Nakajima, A., S. Hashimoto, et al. (2011). Accelerated differentiation of myoblast with high gravitational force in vitro. Biomedical Engineering, 2011 10th International Workshop on.

Nattkemper, T. W., H. J. Ritter, et al. (2001). "A neural classifier enabling high-throughput topological analysis of lymphocytes in tissue sections." Information Technology in Biomedicine, IEEE Transactions on **5**(2): 138-149.

Nattkemper, T. W., T. Twellmann, et al. (2003). "Human vs. machine: evaluation of fluorescence micrographs." Computers in Biology and Medicine **33**(1): 31-43.

Nattkemper, T. W., H. Wersing, et al. (2002). "A neural network architecture for automatic segmentation of fluorescence micrographs." Neurocomputing **48**: 357-367.

Nezamoddin, N. K., F. Paul, et al. (2006). A Statistical Thresholding Method for Cell Tracking. Signal Processing and Information Technology, 2006 IEEE International Symposium on.

Nielsen, L. K., G. K. Smyth, et al. (1991). "HEMACYTOMETER CELL COUNT DISTRIBUTIONS - IMPLICATIONS OF NON-POISSON BEHAVIOR." Biotechnology Progress **7**(6): 560-563.

Noordam, J. C., W. van den Broek, et al. (2005). "A new procedure for the modelling and representation of classes in multivariate images." Chemometrics and Intelligent Laboratory Systems **75**(2): 115-126.

Olivo-Marin, J. C. (2002). "Extraction of spots in biological images using multiscale products." Pattern Recognition **35**(9): 1989-1996.

Olivo, J. C. (1996). Automatic detection of spots in biological images by a wavelet-based selective filtering technique. Image Processing, 1996. Proceedings., International Conference on.

Olson, A. C., N. M. Larson, et al. (1980). "CLASSIFICATION OF CULTURED MAMMALIAN-CELLS BY SHAPE-ANALYSIS AND PATTERN-RECOGNITION." Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences **77**(3): 1516-1520.

Or-Tzadikario, S., R. Sopher, et al. (2010). "Quantitative Monitoring of Lipid Accumulation Over Time in Cultured Adipocytes as Function of Culture Conditions: Toward Controlled Adipose Tissue Engineering." Tissue Engineering Part C-Methods **16**(5): 1167-1181.

Orikawa, J. and T. Tanaka (2010). Cell segmentation from phase-contrast images using hybrid watershed and region growing algorithm for genomic drug discovery. SICE Annual Conference 2010, Proceedings of.

Otaki, T. (2000). "Artifact halo reduction in phase contrast microscopy using apodization." Optical Review **7**(2): 119-122.

Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms." Systems, Man and Cybernetics, IEEE Transactions on **9**(1): 62-66.

Paci, M., L. Nanni, et al. (2011). Computer vision for human stem cell derived cardiomyocyte classification: The induced pluripotent vs embryonic stem cell case study. Computing in Cardiology (CinC), 2011.

Padfield, D., J. Rittscher, et al. (2008). Spatio-temporal cell segmentation and tracking for automated screening. Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on.

Padfield, D., J. Rittscher, et al. (2011). "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis." Medical Image Analysis **15**(4): 650-668.

Paduano, V., L. Sepe, et al. (2010). Time-lapse phase-contrast microscopy fibroblast automated tracking. Imaging Systems and Techniques (IST), 2010 IEEE International Conference on.

Parent, V. A., Garnier, A, Tremblay, JP (2009). Culture medium for myoblasts, precursors thereof and derivatives thereof. Canada. **Patent PCT/CA2009/001342**.

Polerecky, L., A. Bissett, et al. (2009). "Modular Spectral Imaging System for Discrimination of Pigments in Cells and Microbial Communities." Applied and Environmental Microbiology **75**(3): 758-771.

Polzer, H., F. Haasters, et al. (2010). "Quantification of Fluorescence Intensity of Labeled Human Mesenchymal Stem Cells and Cell Counting of Unlabeled Cells in Phase-Contrast Imaging: An Open-Source-Based Algorithm." Tissue Engineering Part C-Methods **16**(6): 1277-1285.

Prats-Montalban, J. M., A. de Juan, et al. (2011). "Multivariate image analysis: A review with applications." Chemometrics and Intelligent Laboratory Systems **107**(1): 1-23.

Prats-Montalban, J. M. and A. Ferrer (2007). "Integration of colour and textural information in multivariate image analysis: defect detection and classification issues." Journal of Chemometrics **21**(1-2): 10-23.

Qian, X., C. Peng, et al. (2009). "Self-organizing map-based multi-thresholding on neural stem cells images." Medical & Biological Engineering & Computing **47**(7): 801-808.
Quanli, W. (2010). A Novel Immersion Simulation Based Self-Organizing Transform with Application to Single-Cell Segmentation from Microscopy Images.
Rajamuthiah, R., B. Fuchs, et al. (2014). "Whole Animal Automated Platform for Drug Discovery against Multi-Drug Resistant Staphylococcus aureus." Plos One **9**(2): e89189.
Rapoport, D. H., T. Becker, et al. (2011). "A Novel Validation Algorithm Allows for Automated Cell Tracking and the Extraction of Biologically Meaningful Parameters." Plos One **6**(11).
Ridler, T. W. and S. Calvard (1978). "Picture Thresholding Using an Iterative Selection Method." Systems, Man and Cybernetics, IEEE Transactions on **8**(8): 630-632.
Rieger, M. A., P. S. Hoppe, et al. (2009). "Hematopoietic Cytokines Can Instruct Lineage Choice." Science **325**(5937): 217-218.
Rimon, N. and M. Schuldiner (2011). "Getting the whole picture: combining throughput with content in microscopy." Journal of Cell Science **124**(22): 3743-3751.
Rodenacker, K. and E. Bengtsson (2003). "A feature set for cytometry on digitized microscopic images." Analytical Cellular Pathology **25**(1): 1-36.
Ross, M. H. and W. Pawlina (2006). Histology, Lippincott Williams & Wilkins.
Rudnicki, M. A., F. Le Grand, et al. (2008). "The Molecular Regulation of Muscle Stem Cell Function." Cold Spring Harbor Symposia on Quantitative Biology **73**: 323-331.
Russell, C., D. Metaxas, et al. (2007). Using the Pⁿ Potts model with learning methods to segment live cell images. Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.
Saad, N. M., S. A. R. Abu-Bakar, et al. (2010). Automated segmentation of brain lesion based on diffusion-weighted MRI using a split and merge approach. Biomedical Engineering and Sciences (IECBES), 2010 IEEE EMBS Conference on.
Sacan, A., H. Ferhatosmanoglu, et al. (2008). "CellTrack: an open-source software for cell tracking and motility analysis." Bioinformatics **24**(14): 1647-1649.
Saha, B. N., N. Ray, et al. (2009). "Snake Validation: A PCA-Based Outlier Detection Method." Signal Processing Letters, IEEE **16**(6): 549-552.
Said, A. F., L. J. Karam, et al. (2007). MIGRATION AND PROLIFERATION ANALYSIS FOR BLADDER CANCER CELLS. Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on.
Sakuma, S. and T. Tanaka (2011). Cell extraction method from phase contrast microscopy image. SICE Annual Conference (SICE), 2011 Proceedings of.
Sanchez, W. Y., T. W. Prow, et al. (2010). "Analysis of the metabolic deterioration of ex vivo skin from ischemic necrosis through the imaging of intracellular NAD(P)H by multiphoton tomography and fluorescence lifetime imaging microscopy." Journal of Biomedical Optics **15**(4): 046008.
Satulovsky, J., R. Lui, et al. (2008). "Exploring the control circuit of cell migration by mathematical Modeling." Biophysical Journal **94**(9): 3671-3683.
Schilling, T., L. Miroslaw, et al. (2007). "Towards rapid cervical cancer diagnosis: automated detection and classification of pathologic cells in phase-contrast images." International Journal of Gynecological Cancer **17**(1): 118-126.
Selinummi, J., P. Ruusuvuori, et al. (2009). "Bright Field Microscopy as an Alternative to Whole Cell Fluorescence in Automated Analysis of Macrophage Images." Plos One **4**(10).
Sethian, J. A. and D. Adalsteinsson (1997). "An overview of level set methods for etching, deposition, and lithography development." Ieee Transactions on Semiconductor Manufacturing **10**(1): 167-184.

Seungil, H., D. F. E. Ker, et al. (2011). "Automated Mitosis Detection of Stem Cell Populations in Phase-Contrast Microscopy Images." Medical Imaging, IEEE Transactions on **30**(3): 586-596.

Shaner, N. C., P. A. Steinbach, et al. (2005). "A guide to choosing fluorescent proteins." Nat Meth **2**(12): 905-909.

Shen, L. L. and L. Bai (2006). "A review on Gabor wavelets for face recognition." Pattern Analysis and Applications **9**(2-3): 273-292.

Shenkman, B. S., O. V. Turtikova, et al. (2010). "Skeletal muscle activity and the fate of myonuclei." Actanaturae **2**(2): 59-66.

Shuler, M. L. and F. Kargi (1992). Bioprocess Engineering: Basic Concepts. Englewood Cliffs, New Jersey.

Siddiqi, A. M., H. Li, et al. (2008). "Use of hyperspectral imaging to distinguish normal, precancerous, and cancerous cells." Cancer Cytopathology **114**(1): 13-21.

Sigoillot, F. D., J. F. Huckins, et al. (2011). "A Time-Series Method for Automated Measurement of Changes in Mitotic and Interphase Duration from Time-Lapse Movies." Plos One **6**(9).

Sirovich, L. and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces." J. Opt. Soc. Am. A **4**(3): 519-524.

Skoczylas, M., W. Rakowski, et al. (2011). "Unstained viable cell recognition in phase-contrast microscopy." Opto-Electronics Review **19**(3): 307-319.

Skoczylas, M., W. Rakowski, et al. (2011). "WAVELET-SVM CLASSIFICATION AND AUTOMATIC RECOGNITION OF UNSTAINED VIABLE CELLS IN PHASE-CONTRAST MICROSCOPY." Radiation Protection Dosimetry **143**(2-4): 353-357.

Skuk, D., M. Goulet, et al. (2007). "First test of a “high-density injection” protocol for myogenic cell transplantation throughout large volumes of muscles in a Duchenne muscular dystrophy patient: eighteen months follow-up." Neuromuscular disorders **17**(1): 38-46.

Sommer, C. and D. Gerlich (2013). "Machine learning in cell biology - teaching computers to recognize phenotypes." Journal of Cell Science **126**(Pt 24): 5529-5539.

Starck, J. L. and A. Bijaoui (1994). "FILTERING AND DECONVOLUTION BY THE WAVELET TRANSFORM." Signal Processing **35**(3): 195-211.

Starck, J. L., E. J. Candes, et al. (2002). "The curvelet transform for image denoising." Ieee Transactions on Image Processing **11**(6): 670-684.

Starck, J. L., M. Elad, et al. (2004). "Redundant multiscale transforms and their application for morphological component separation." Advances in Imaging and Electron Physics, Vol 132 **132**: 287-348.

Sternberg, S. R. (1983). "Biomedical Image Processing." Computer **16**(1): 22-34.

Sulaiman, S. N., N. A. M. Isa, et al. (2010). Pseudo Color Features Extraction technique for cervical cancer of Pap smear images. Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on.

Sulaiman, S. N., N. A. M. Isa, et al. (2010). Overlapping cells separation method for cervical cell images. Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on.

Sungeun, E., R. Bise, et al. (2010). Detection of hematopoietic stem cells in microscopy images using a bank of ring filters. Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on.

Tabatabai, M., D. Williams, et al. (2005). "Hyperbolastic growth models: theory and application." Theoretical Biology and Medical Modelling **2**(1): 14.

Tabatabai, M. A., Z. Bursac, et al. (2011). "Mathematical modeling of stem cell proliferation." Medical & Biological Engineering & Computing **49**(3): 253-262.

Taoyi, C., Z. Yong, et al. (2010). Neural stem cell segmentation using local complex phase information. Image Processing (ICIP), 2010 17th IEEE International Conference on.

Tessier, J. (2006). Détermination de la composition de l'alimentation des circuits de broyage par analyse d'images multivariée. Département de génie chimique. Québec, Canada, Université Laval. **Maître ès sciences (M.Sc.):** 107p.

Theriault, D. H., M. L. Walker, et al. (2012). "Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning." Machine Vision and Applications **23**(4): 659-673.

Topman, G., O. Sharabani-Yosef, et al. (2011). "A Method for Quick, Low-Cost Automated Confluency Measurements." Microscopy and Microanalysis **17**(06): 915-922.

Tsien, R. Y. (2009). "Constructing and Exploiting the Fluorescent Protein Paintbox (Nobel Lecture)." Angewandte Chemie-International Edition **48**(31): 5612-5626.

Tuceryan, M. and A. K. Jain (1999). Texture Analysis. Handbook of Pattern Recognition and Computer Vision (2nd Edition). C. H. Chen, L. F. Pau, C. C. L. F. P. P. S. Wang and P. S. P. Wang, World Scientific.

Turk, M. (1991). "Eigenfaces for recognition." Journal of cognitive neuroscience(3.1): 71-86.

Turk, M. A. and A. P. Pentland (1991). Face recognition using eigenfaces. Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on.

Umetrics (2005). SIMCA-P and SIMCA-P+ 11: User Guide and Tutorial. Umea.

Usenik, P., T. Vrtovec, et al. (2011). "Automated tracking and analysis of phospholipid vesicle contours in phase contrast microscopy images." Medical and Biological Engineering and Computing **49**(8): 957-966.

Vallotton, P. and S. Olivier (2013). "Tri-track: Free Software for Large-Scale Particle Tracking." Microscopy and Microanalysis **19**(02): 451-460.

Vincent, L. and P. Soille (1991). "Watersheds in digital spaces: an efficient algorithm based on immersion simulations." Pattern Analysis and Machine Intelligence, IEEE Transactions on **13**(6): 583-598.

Vindin, H., L. Bischof, et al. (2014). "Validation of an algorithm to quantify changes in actin cytoskeletal organization." Journal of biomolecular screening **19**(3): 354-368.

Vogels, M., R. Zoeckler, et al. (1975). "P. F. Verhulst's “notice sur la loi que la populations suit dans son accroissement” from correspondence mathematique et physique. Ghent, vol. X, 1838." Journal of Biological Physics **3**(4): 183-192.

Walck, C. (1996). Hand-book on STATISTICAL DISTRIBUTIONS for experimentalists.

Walker, R. F., P. Jackway, et al. (1994). Classification of cervical cell nuclei using morphological segmentation and textural feature extraction. Intelligent Information Systems,1994. Proceedings of the 1994 Second Australian and New Zealand Conference on.

Weber, S., M. L. Fernandez-Cachon, et al. (2013). "Label-Free Detection of Neuronal Differentiation in Cell Populations Using High-Throughput Live-Cell Imaging of PC12 Cells." Plos One **8**(2).

Widengren, J. and R. Rigler (1996). "Mechanisms of photobleaching investigated by fluorescence correlation spectroscopy." Bioimaging **4**(3): 149-157.

Wold, S. (1978). "Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models." Technometrics **20**(4): 397-405.

Wold, S., K. Esbensen, et al. (1987). "PRINCIPAL COMPONENT ANALYSIS." Chemometrics and Intelligent Laboratory Systems **2**(1-3): 37-52.

Wold, S., P. Geladi, et al. (1987). "Multi-way principal components-and PLS-analysis." Journal of Chemometrics **1**(1): 41-56.

Wold, S., M. Sjostrom, et al. (2001). "PLS-regression: a basic tool of chemometrics." Chemometrics and Intelligent Laboratory Systems **58**(2): 109-130.

Wu, C. H., J. Schulte, et al. (2010). "Automatic Robust Neurite Detection and Morphological Analysis of Neuronal Cell Cultures in High-content Screening." Neuroinformatics **8**(2): 83-100.

Xiang, Q. and Y. Datian (2009). Automatic Segmenting and Classifying the Neural Stem Cells in Adherent Culturing Condition. Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on.

Xiaodong, Y., L. Houqiang, et al. (2006). "Nuclei Segmentation Using Marker-Controlled Watershed, Tracking Using Mean-Shift, and Kalman Filter in Time-Lapse Microscopy." Circuits and Systems I: Regular Papers, IEEE Transactions on **53**(11): 2405-2414.

Xiong, W., S.-C. Chia, et al. (2011). "Detection of unstained living neurospheres from phase contrast images with very large illumination variations." Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference **2011**: 6154-6157.

Yin, Z. Z., K. Li, et al. (2010). Understanding the Optics to Aid Microscopy Image Segmentation. Medical Image Computing and Computer-Assisted Intervention - Miccai 2010, Pt I. T. Jiang, N. Navab, J. P. W. Pluim and M. A. Viegever. Berlin, Springer-Verlag Berlin. **6361:** 209-217.

Yu, B. Y., C. Elbuken, et al. (2011). "Image processing and classification algorithm for yeast cell morphology in a microfluidic chip." Journal of Biomedical Optics **16**(6).

Yu, H. L. and J. F. MacGregor (2003). "Multivariate image analysis and regression for prediction of coating content and distribution in the production of snack foods." Chemometrics and Intelligent Laboratory Systems **67**(2): 125-144.

Yu, S., S. Duthaler, et al. (2005). Autofocusing algorithm selection in computer microscopy. Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on.

Zeiss. (2013). "Education in Microscopy and Digital Imaging." from http://zeiss-campus.magnet.fsu.edu/articles/basics/contrast.html.

Zhang, Y. J. (1996). "A survey on evaluation methods for image segmentation." Pattern Recognition **29**(8): 1335-1346.

Zhaozheng, Y. and T. Kanade (2011). Restoring artifact-free microscopy image sequences. Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on.

Zhou, C., G. L. Zhang, et al. (1995). A general evaluation method for segmentation algorithm based on experimental design methodology. New York, I E E E.

Zhou, Y. (2007). CELL SEGMENTATION USING LEVEL SET METHOD. Johann Radon Institute for Computational and Applied Mathematics. Linz, Johannes Kepler Universit¨at. **M. Sc.**

Zimmer, C., E. Labruyere, et al. (2002). "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing." Ieee Transactions on Medical Imaging **21**(10): 1212-1221.

Appendix A. Hyperspectral microscopy

A.1 Introduction

In previous chapters, cell culture experiments involved conventional PCM images acquired using a monochrome camera (Qimaging® camera). However measuring and analyzing the full transmission or emission spectra of the cells growing in a culture medium for each pixel could provide useful chemical information (using NIR spectra for instance) about the cellular samples. Even though hyperspectral microscopy was not the main focus of this PhD project, preliminary tests were performed to assess its potential from a technical perspective for cell characterization. The results are summarized in this chapter so it could provide useful insights for future projects.

A.2 Methodology

A.2.1 Hyperspectral cameras specifications

Three hyperspectral cameras (spectrometer attached to a camera CCD) were available for this study: a ultraviolet (UV), a visible (VIS) and the near-infrared (NIR) hyperspectral camera (shown in Figure A-1). These cameras can be fitted on the IX81 microscope lateral port. The cameras specifications are provided in Table A-1.

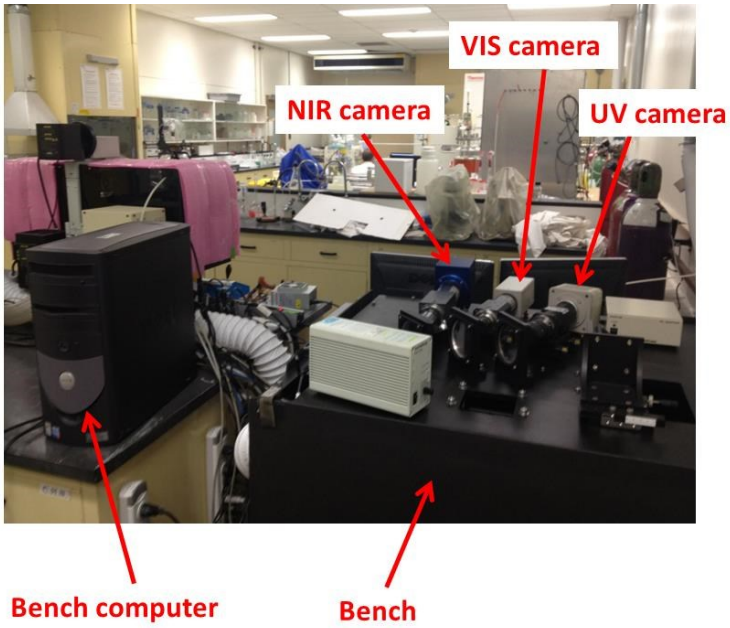


Figure A-1 : UV, VIS and NIR hyperspectral cameras available on the bench (can be fitted on the IX81 microscope lateral port)

Table A-1 : Hyperspectral cameras specifications

Hyperspectral camera name	Camera Model	Spectro Model	Nominal wavelengths (nm)	Drivers Compatibility
UV	Hamamatsu® C8800-01C	ImSpector® UV4E	200 – 400	Not tested
VIS	Hamamatsu® C8484-05G	ImSpector® V10E	400 - 1000	Compatible with Metamorph®, driver (Wasabi®) not compatible with Qimaging® camera drivers (cannot be used at the same time)
NIR	Xenics® XEVA-USB	ImSpector® N17E	900 - 1700	Drivers not compatible with Metamorph®

The spectrometers associated to the hyperspectral cameras disperse wavelengths along the vertical axis of the CCD, and pixels in this direction correspond to light at a certain wavelength. This is the main reason why calibration is required, to associate a pixel position along the vertical axis to a wavelength intensity. A radiometric calibration was performed using a mercury lamp having intensity peaks at known wavelengths to associate each pixel in each slice image (pixel number) along the vertical axis to a given wavelength. The calibration results for the VIS and NIR hyperspectral cameras are provided in Figure A-2. From the linear regressions performed, it is possible to see that the VIS camera has a spectral resolution ($\Delta\lambda$) of 0.61 nm per pixel and the NIR camera has a spectral resolution of 2.9 nm per pixel in the λ axis.

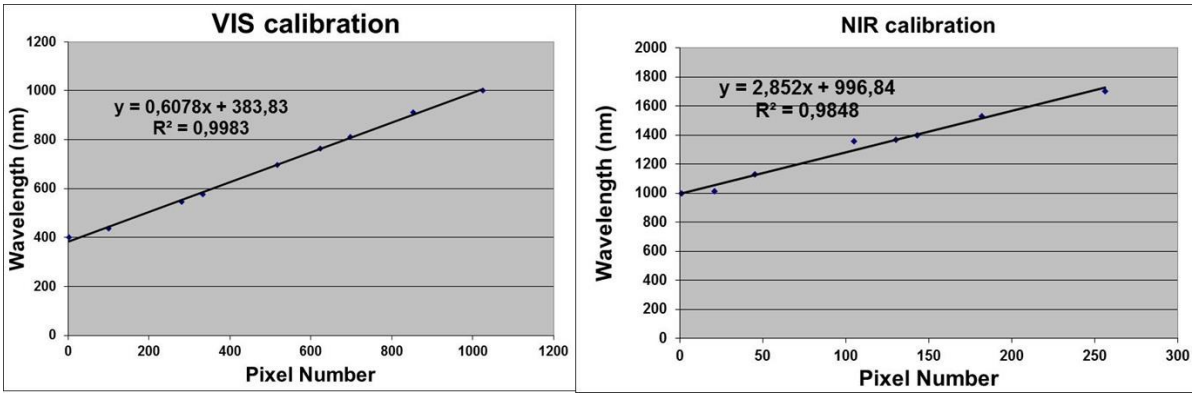


Figure A-2 : VIS and NIR camera calibration

Both the VIS and NIR hyperspectral cameras were used and tested in this project. When the VIS hyperspectral camera was installed on the microscope computer, it was necessary to install the Hamamatsu® camera drivers (Wasabi®) which reinitialized the USB ports, meaning that the Qimaging® camera did not work anymore on the microscope computer. When switching back to the Qimaging® camera, it was necessary to reinstall the Qimaging® camera drivers to reinitialize the USB ports, meaning that the VIS hyperspectral camera did not work anymore on the microscope computer. Even though the Qimaging® and the Hamamatsu® camera drivers are not compatible, both the Qimaging® and the VIS hyperspectral cameras can be controlled using Metamorph® for image acquisition (however not at the same time).

Since Xenics® cameras cannot be controlled by Metamorph®, when using the NIR hyperspectral camera, the microscope and the stage were controlled by Metamorph® (on the microscope computer) while images were acquired in streaming by LabView® on the bench computer. A mapping algorithm was developed in Matlab® (see Appendix G. for HYPERSPECTRAL_MIA_v0r3.m) to relate each dummy frame saved by Metamorph® to the corresponding frame acquired by LabView®.

A.2.2 Control of automated stage and hyperspectral cameras

A push broom set-up was used to acquire whole hyperspectral cubes (slices in y and λ dimensions acquired at fixed x intervals, as illustrated in Figure 2-19), and the main challenge was to control at the same time the automated stage, the hyperspectral camera acquisition and the different functions of the microscope (light source shutter, filter cubes for epifluorescence, etc.). Journals were programmed in Metamorph® to acquire hyperspectral cubes in a similar fashion as described before (while controlling the stage

and the microscope), but with specific lists of points that were generated with a Matlab® script (**POINTS_GENERATOR_METAMORPH_v0r1.m**) provided in Appendix G. To generate lists of points for hyperspectral microscopy, Δx and Δy were estimated using the method described in Section A.2.3.

A.2.3 Estimation of spatial resolutions Δx and Δy

The spatial resolution can be defined as the size of the pixels (in µm) in the x and y directions. The spatial resolution achievable is influenced by the microscope optics and by the cameras CCD and optical path. An estimation of the spatial resolutions Δx and Δy of the images acquired by the spectral cameras was obtained by imaging a hemacytometer, and by adjusting Δx by trial-and-error until the grid inside the hemacytometer appeared square. After each trial, a $\mathbf{T_1}$ score image was computed (see **T1_generation.m**, Appendix G.) from the hypercube to assess whether the results were acceptable. An example of a brightfield hyperspectral $\mathbf{T_1}$ score image of a hemacytometer imaged at 10X is provided in Figure A-3. Note that since this adjustement was performed by trial-and-error, the hemacytometer lines are not perfectly square, meaning that the Δx estimation might be slightly biased (this adjustment was a long process).

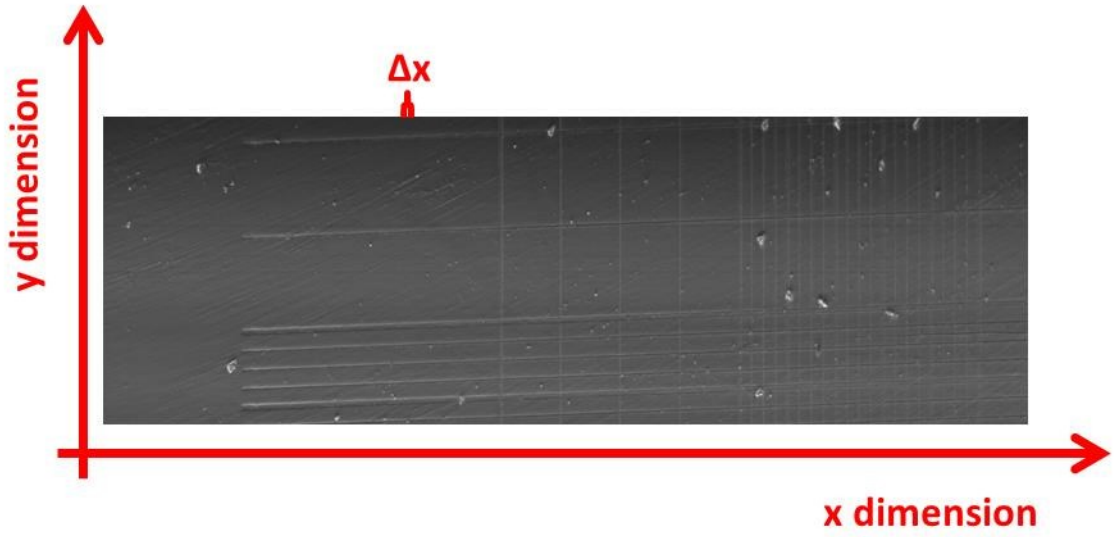


Figure A-3 : Brightfield hyperspectral T_1 score image of a hemacytometer at 10X, with Δx = 1 µm between slices

Estimations of Δx and Δy for the VIS hyperspectral camera are provided in Table A-2:

Table A-2 : Estimated Δx and Δy for VIS hyperspectral microscopy

Magnification	Δx (µm)	Δy (µm)
10X	1.0	0.61
20X	0.4	0.31
40X	0.2	0.15

A.2.4 Light sources

Two halogen light sources (similar casing) can be fixed onto the microscope: the one that was originally provided with the IX81 microscope ("Microscope light source", model U-LH100-3, Olympus®) and the one installed on the hyperspectral bench behind the microscope ("Bench light source", model U-LH100-3, Olympus®, used by Gosselin (2010)). The Microscope light source was tested by the COPL (Centre d'optique, photonique et laser, Université Laval), and the analysis showed that wavelengths higher than 750 nm were blocked by a filter integrated in the light source (see Figure A-4). It can be hypothesized that the purpose of this filter is to prevent cell cultures from overheating when exposed to near-infrared light. However the Bench light source does not have this filter, as can be seen in Figure A-5.

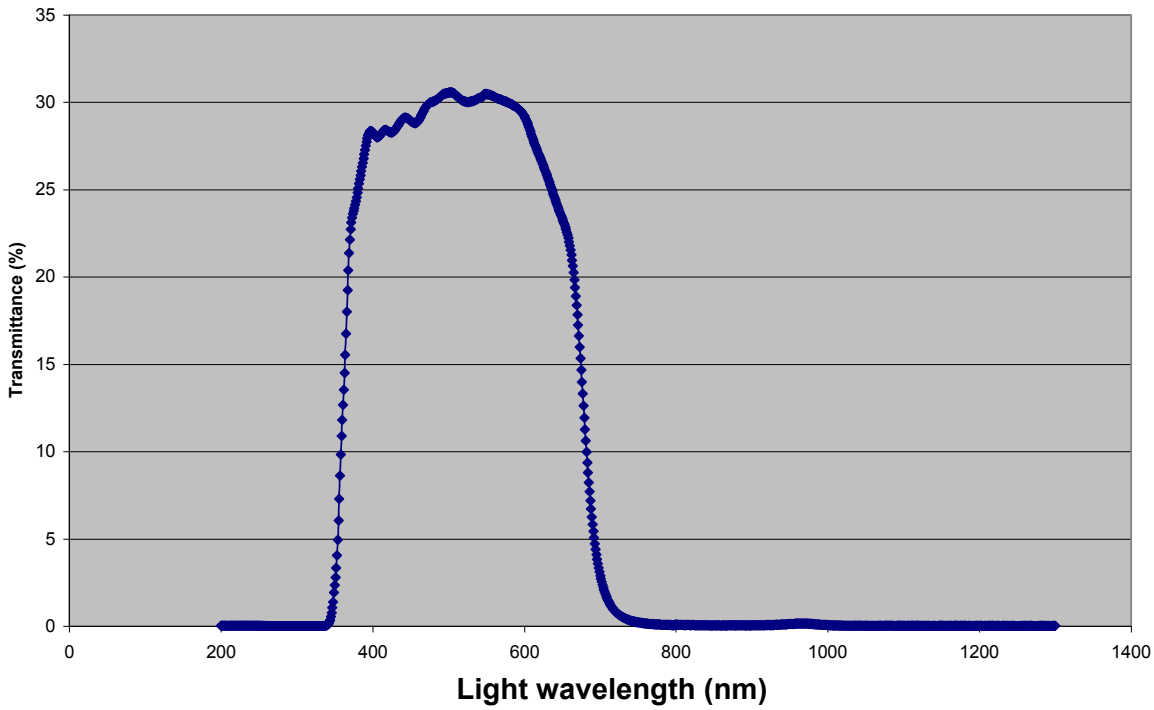


Figure A-4 : Transmittance as a function of light wavelength (in nm) for the Microscope light source tested by the COPL. Wavelengths higher than 750 nm are blocked.

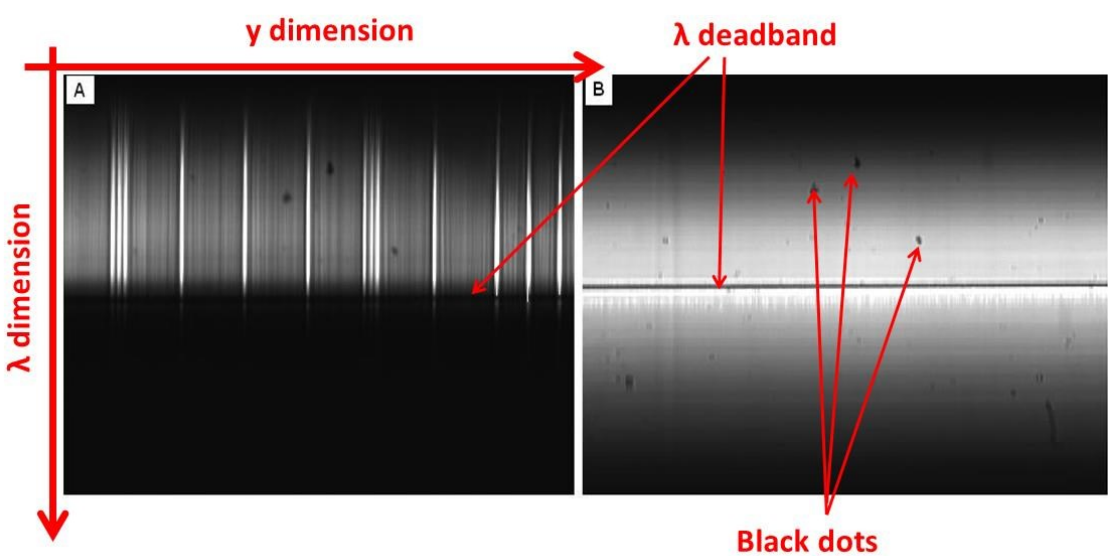


Figure A-5 : λ and y slices images acquired through the IX81 Microscope with A-Microscope light source (imaging a hemacytometer); B-Bench light source. The bench light source does not block wavelengths at 750 nm.

Since the bench light source does not have a filter, it was used for hyperspectral microscopy experiments. However there are still several issues to solve in future work, since it is possible to see in Figure A-5B that there is a wavelength deadband (around λ =700 nm) that remains black and since there are several black dots that are localized both spatially and spectrally. In both cases, the light is most likely filtered somewhere in the microscope optical patch.

A.3 Experiments description

A series of experiments was performed at the end of 2012 and at beginning of 2013 (in 6 well multiwell plates with a magnification of 20X) to see the potential of hyperspectral microscopy for cell characterization. A total of 5 experiments were conducted and are detailed in Table A-3. The experiment started on November 8th 2012 was performed over several days (from seeding to confluence) in live-cell imaging as the other experiments were only performed once the cells achieved confluence (one acquisition for each well at the end of the cell culture). The goal was to see how the cells could be segmented and characterized by performing MIA on the measured hyperspectral cubes and also to verify if myoblasts grown in SSM and SFM could be distinguished based on spectral data.

Table A-3 : Series of experiments realized in hyperspectral microscopy

Experiment Name	Cell culture start date	Cells	Medium	Cell initial concentration (c/mL)	Hyperspectral cameras used	Description
Experiment_November_08_2012	2012-11-07	BB13MP8	Well 1: SSM+cells Well 2: SSM Well 3: SFM+cells Well 4: SFM	40000	VIS	Imaging wells over several days for hyperspectral live-cell imaging (PCM-VIS)
Experiment_January_05_2013	2012-12-30	BB13MP4	Well 1: SSM+cells Well 2: SSM+cells Well 3: SSM Well 4: SFM+cells Well 5: SFM+cells Well 6: SFM	11250	VIS	Imaging once in PCM-VIS and EM-VIS
Experiment_January_06_2013	2013-01-03	BB13MP5	Well 1: SFM+cells Well 2: SFM+cells Well 3: SFM Well 4: SSM+cells Well 5: SSM+cells Well 6: SSM	155156	VIS and NIR	Imaging once in PCM-VIS, EM-VIS and PCM-NIR
Experiment_January_11_2013	2013-01-06	BB13MP6	Well 1: SSM+cells Well 2: SSM+cells Well 3: SSM Well 4: SFM+cells Well 5: SFM+cells Well 6: SFM	25782	VIS and NIR	Imaging once in PCM-VIS, EM-VIS and PCM-NIR
Experiment_January_16_2013	2013-01-10	BB13MP7	Well 1: SFM+cells Well 2: SFM+cells Well 3: SFM Well 4: SSM+cells Well 5: SSM+cells Well 6: SSM	10938	VIS and NIR	Imaging once in PCM-VIS, EM-VIS and PCM-NIR

Adjusting the focus for each well was a challenge when acquiring hyperspectral cubes. The focus was adjusted manually for each well (see Section E.6), but without using the autofocus utility. A Δx of 0.35 μm was used for the slices, and the cubes consist of 1001 contiguous slices forming hypercubes of roughly 350 μm in length (in x) and of 413 μm in width (in y). Note that the spatial resolution of the NIR camera is most likely different from the resolution of the VIS camera. However since it was difficult to assess it was assumed to be roughly the same (as these were preliminary tests). Hypercubes generated from the VIS camera have a length of 1001 pixels (x axis), a width of 1344 pixels (y axis) and a height of 1024 pixels (λ axis). Hypercubes generated from the NIR camera have a length of 1001 pixels (x axis), a width of 320 pixels (y axis) and a height of 256 pixels (λ axis).

A.4 Phase contrast hyperspectral microscopy with VIS camera (PCM-VIS)

For each experiment (January 5th, 6th, 11th and 16th 2013), PCM-VIS hyperspectral microscopy cubes captured for each well were analyzed by performing a PCA on all available hyperspectral image (i.e. a single PCA model built using the kernel MIA algorithm). All the experiments analyzed gave similar results, so only those of the January 6th 2013 experiment are presented. The loadings, the score images and the score density histograms for this particular experiment are shown in Figure A-6, Figure A-7, Figure A-8, Figure A-9 and Figure A-10. To verify if culture time has an influence on the results, a second PCA model was built on the PCM-VIS hyperspectral microscopy cubes for the experiment of November 8th 2012 only (for each well and for each time acquisition). The PCA analysis provided a similar latent space, and typical $\mathbf{T}_1$ score results are provided in Figure A-11.

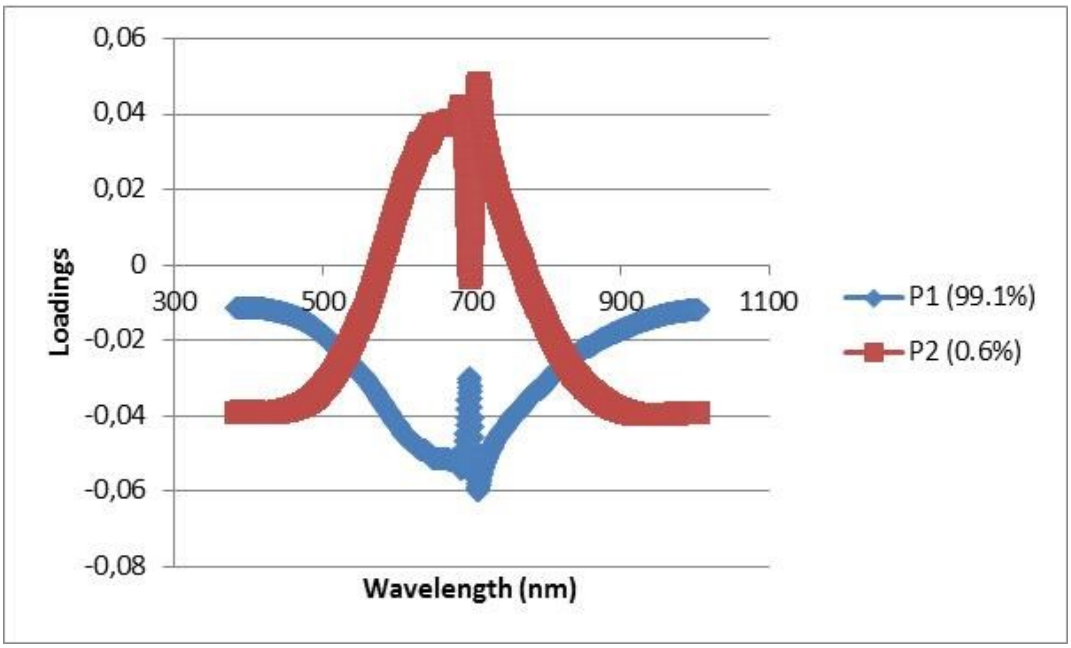


Figure A-6 : p_1 and p_2 loadings from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013

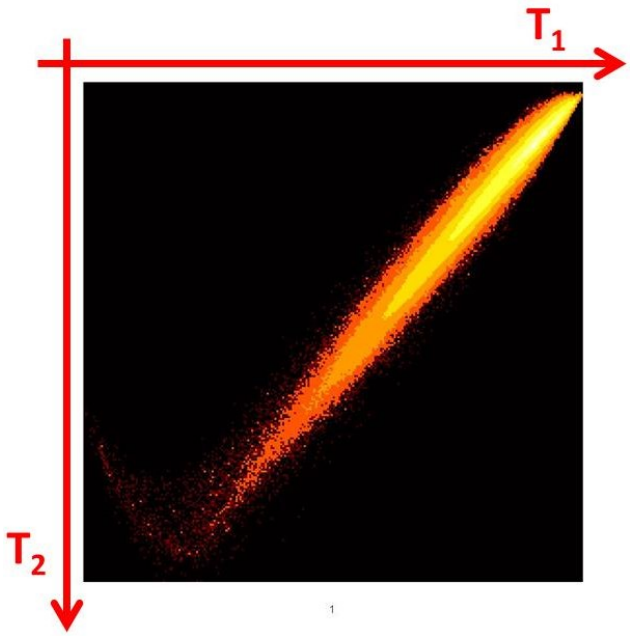


Figure A-7 : T_1/T_2 global score density histogram from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013

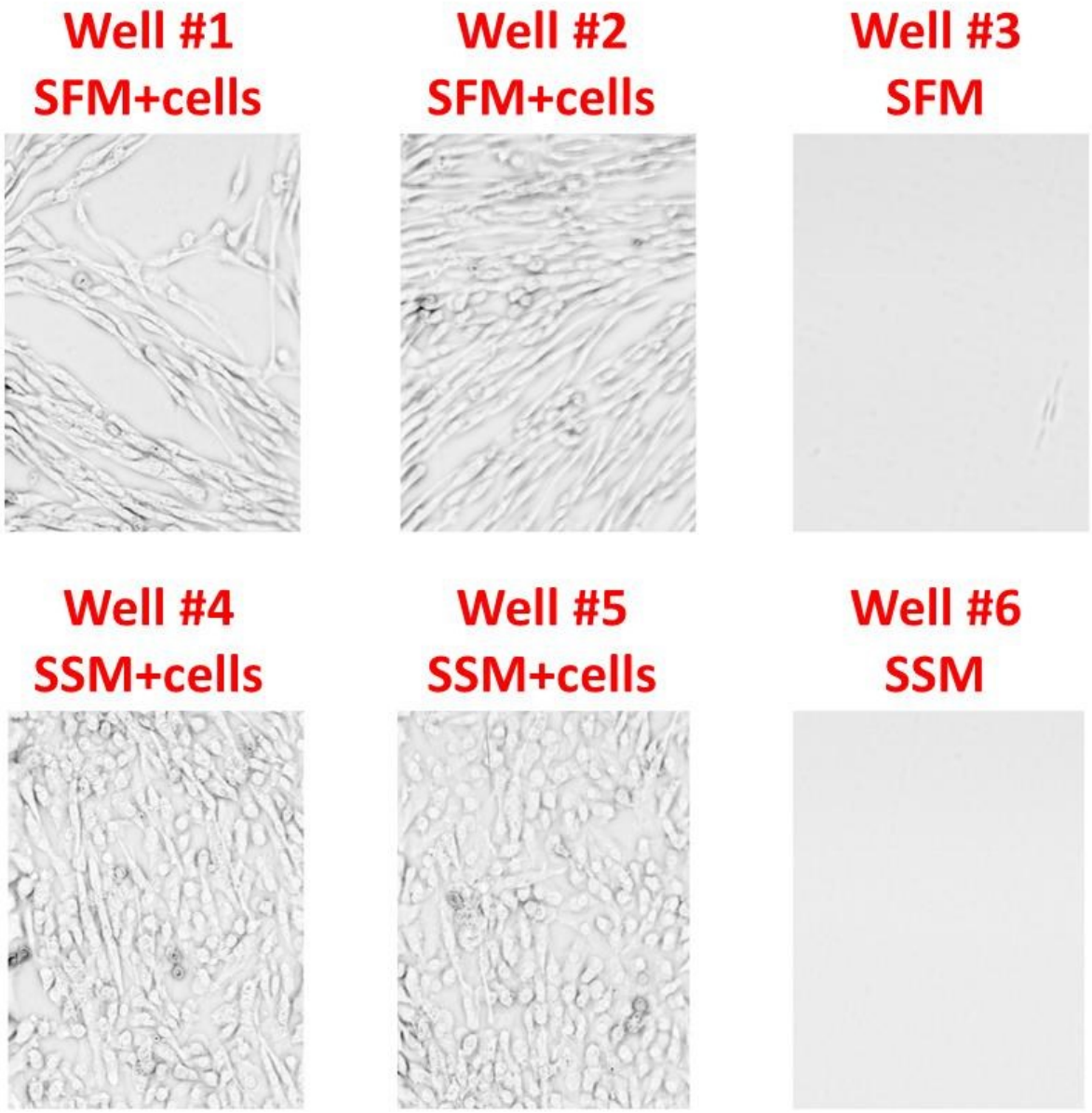


Figure A-8 : T_1 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013

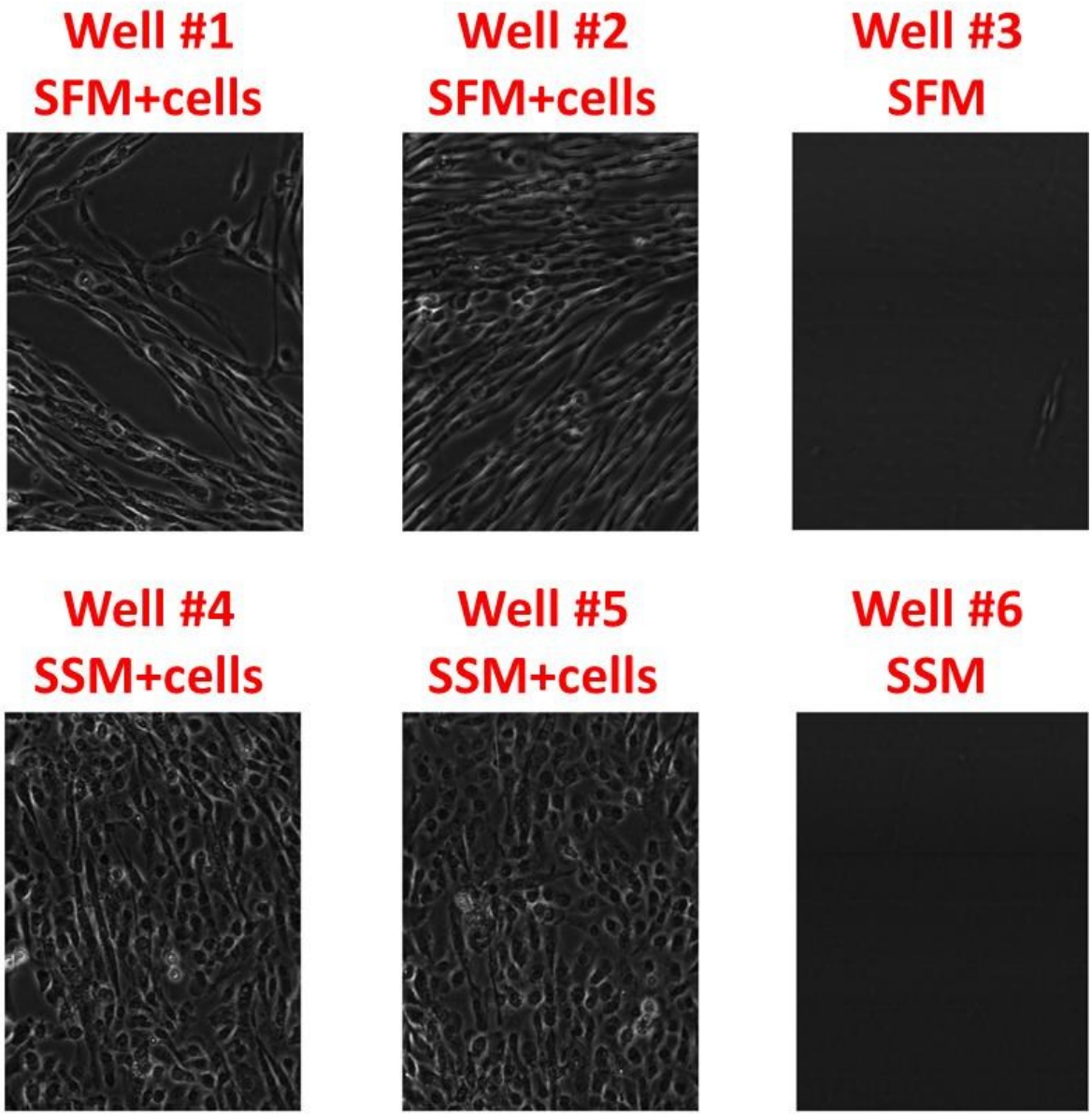


Figure A-9 : T_2 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013

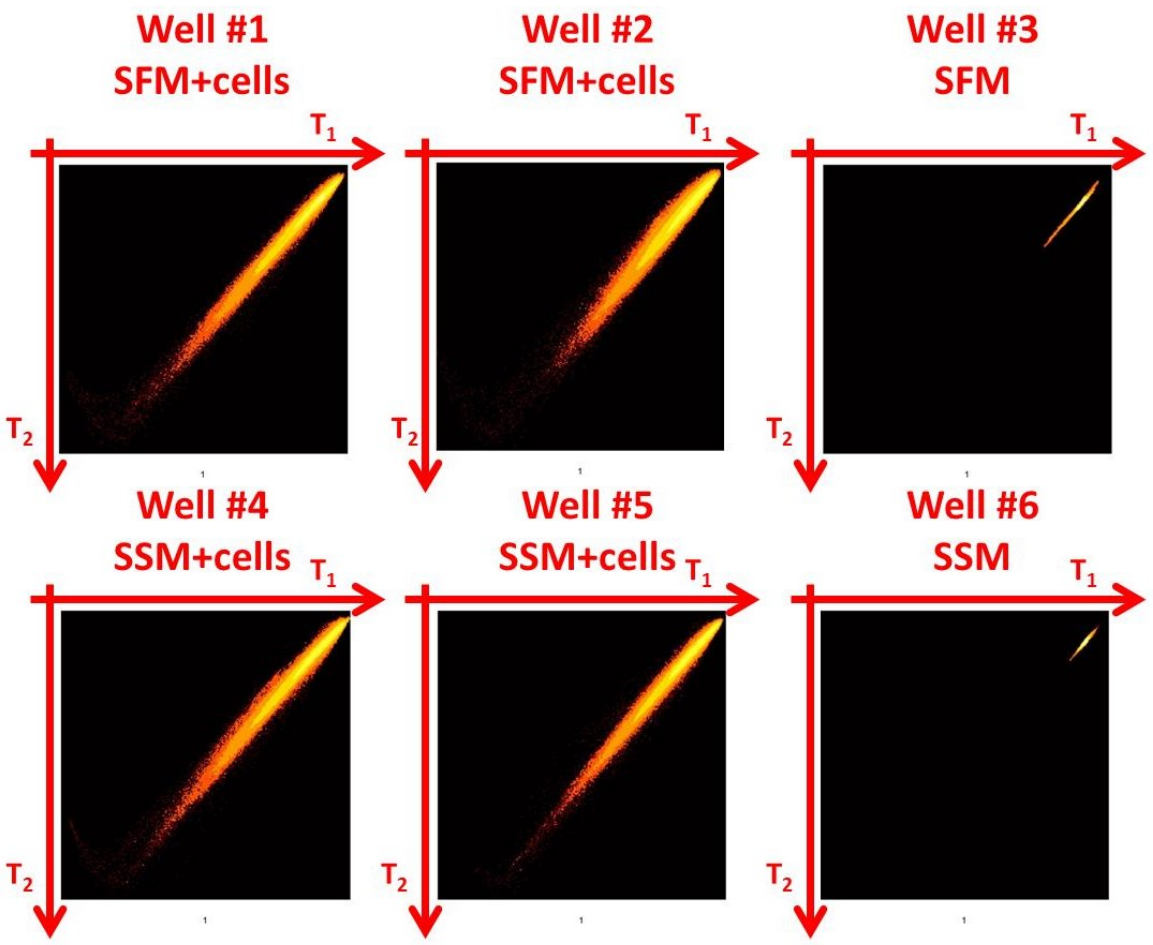


Figure A-10 : T_1/T_2 individual score density histograms from PCA performed on PCM-VIS hyperspectral cubes for experiment January 6th 2013

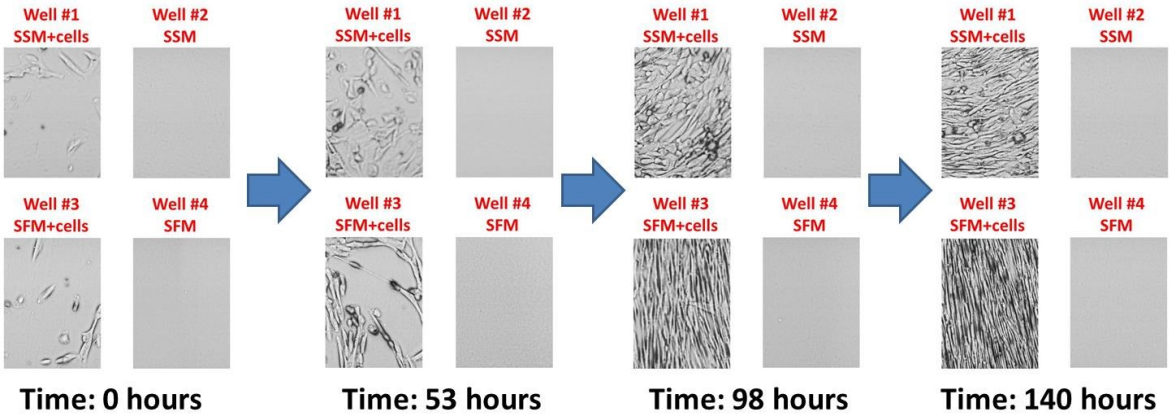


Figure A-11 : T_1 score images from PCA performed on PCM-VIS hyperspectral cubes for experiment November 8th 2012

In Figure A-6, it is possible to see a discrepancy zone in the loadings around 700 nm, which is due to the λ deadband (see Figure A-5). The loadings follow the bell shape of the spectrum associated to the bench light source. Looking at the $\mathbf{T_1}/\mathbf{T_2}$ score density histograms (Figure A-7 and Figure A-10), it appears that most of the wavelengths are strongly correlated with each other (almost a straight line). The main variation is due to light intensity variations related to PCM.

A.5 Epifluorescence hyperspectral microscopy with VIS camera (EM-VIS)

For each experiment (January 5th, 6th, 11th and 16th 2013), epifluorescence hyperspectral microscopy cubes associated to each well were analyzed through an overall PCA (one PCA per experiment, using a kernel MIA algorithm). The goal was to measure the autofluorescence of the cells. All the experiments analyzed gave similar results, so only the results associated to the experiment of January 6th 2013 are presented. The loadings, the score images and the score density histograms for the analysis of the experiment ran on January 6th 2013 are shown in Figure A-12, Figure A-13, Figure A-14, Figure A-15 and Figure A-16.

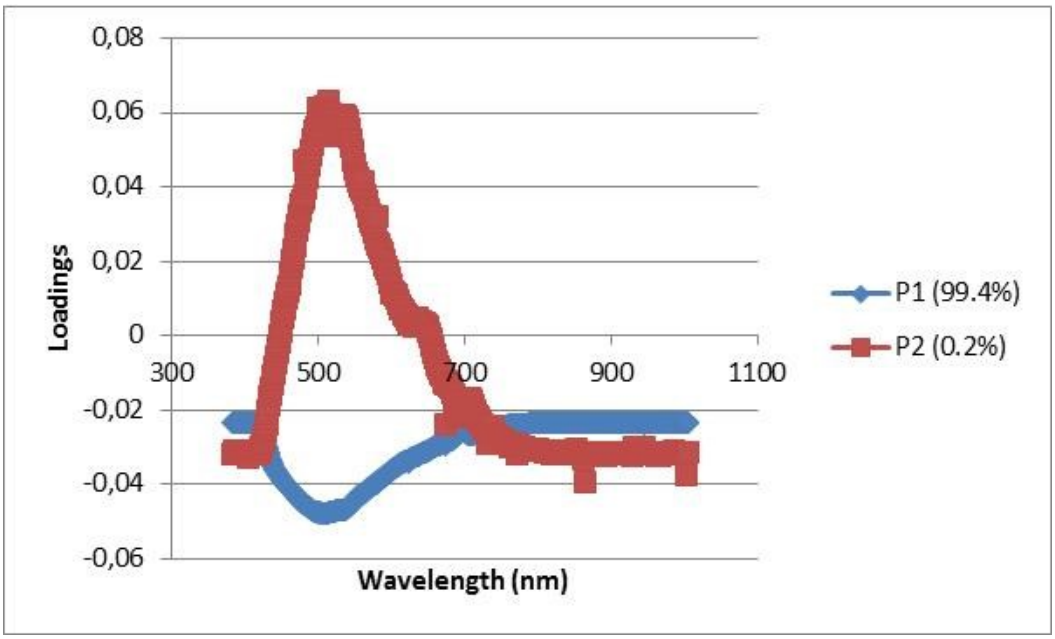


Figure A-12 : p_1 and p_2 loadings from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013

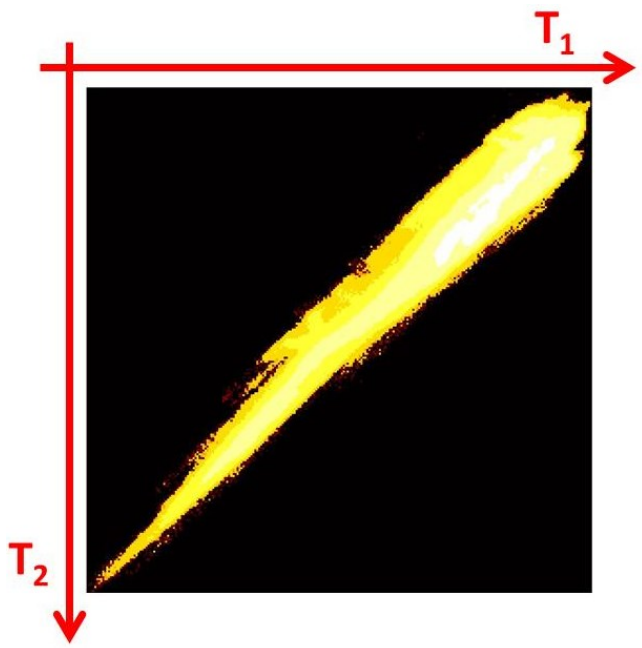


Figure A-13 : T_1/T_2 global score density histogram from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013

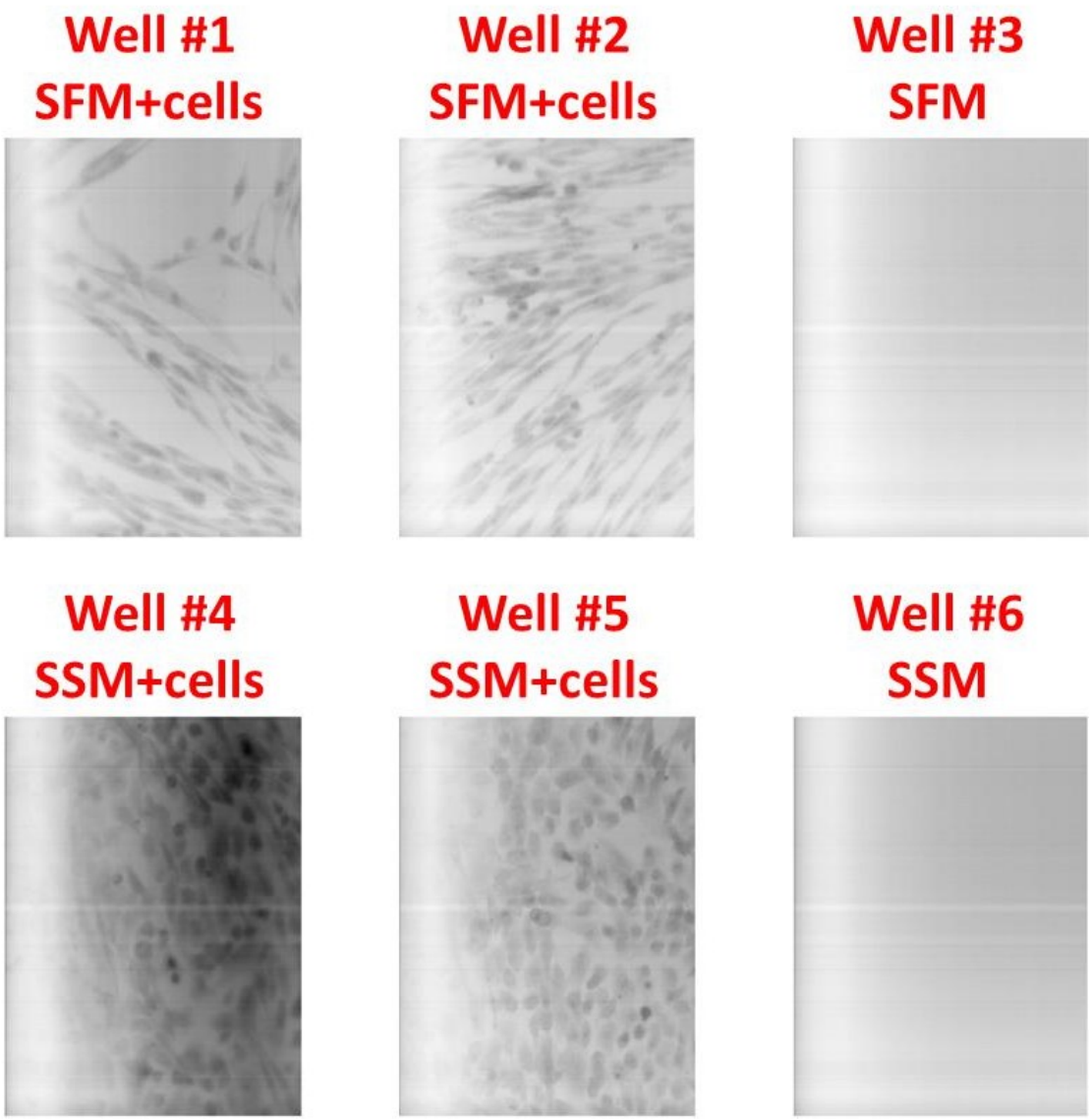


Figure A-14 : T_1 score images from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013

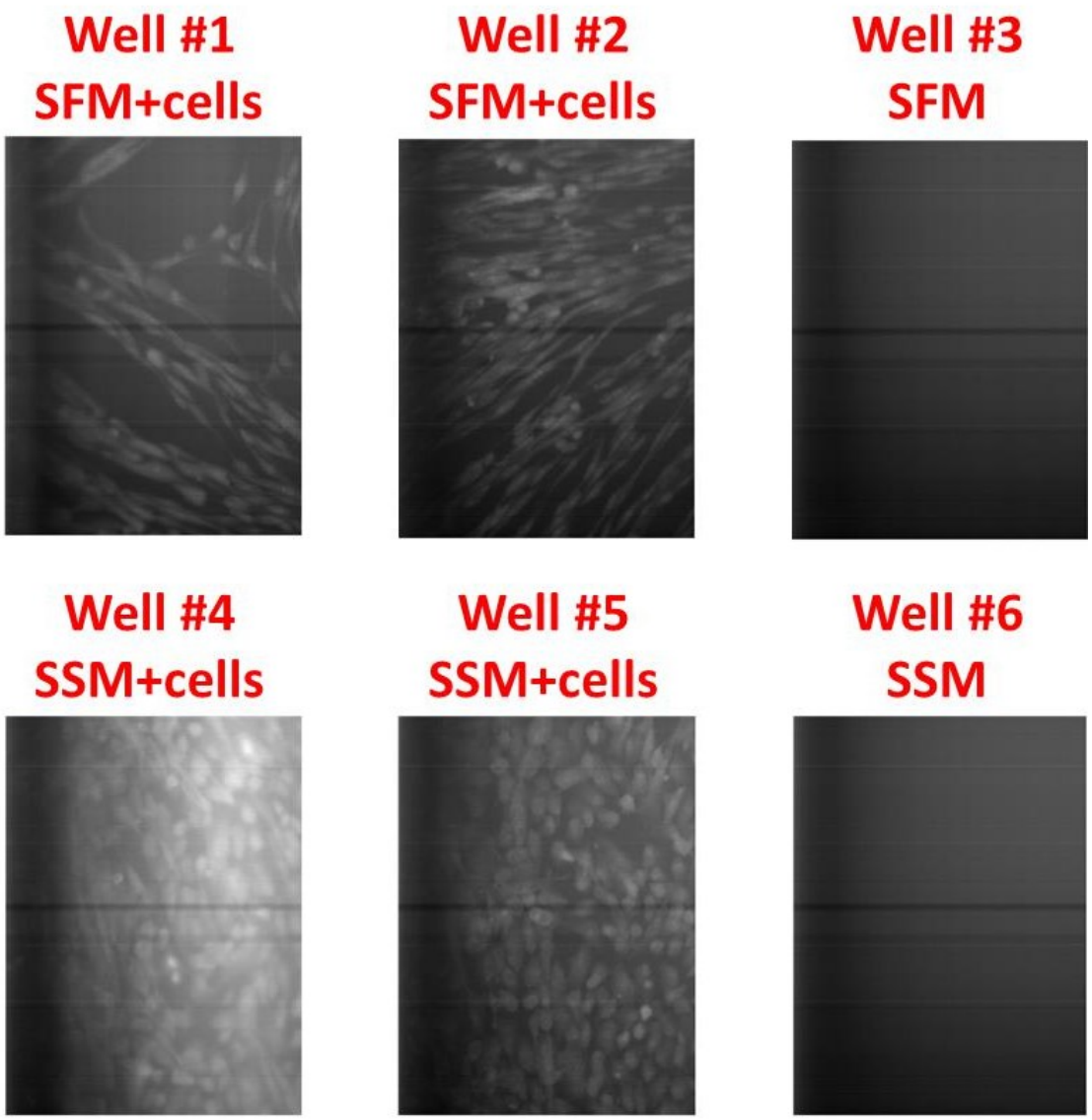


Figure A-15 : T_2 score images from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013

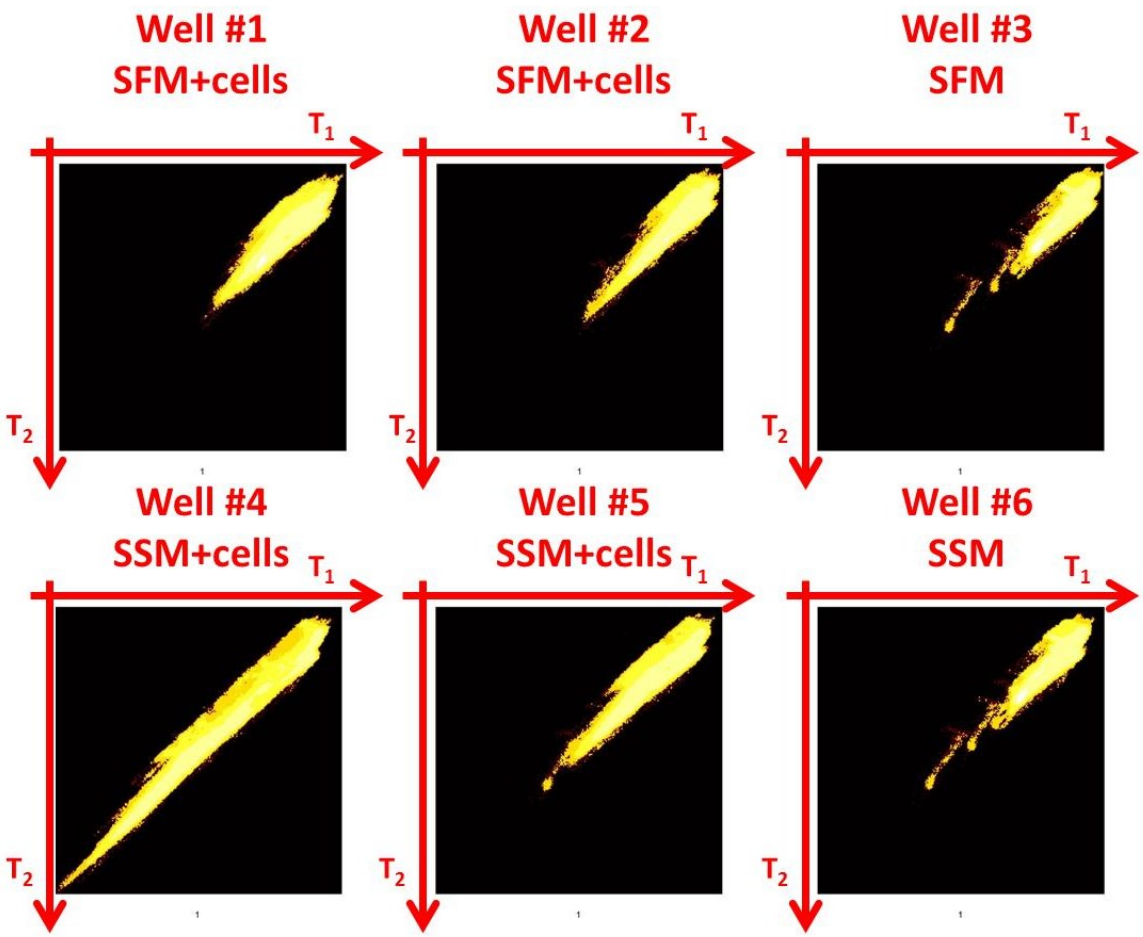


Figure A-16 : T_1/T_2 individual score density histograms from PCA performed on EM-VIS hyperspectral cubes for experiment January 6th 2013

The first component captures 99.4% of the variance as the second component captures 0.2% of the variance. The loadings of the first and second components ($\mathbf{p}_1$ and $\mathbf{p}_2$ in Figure A-12) capture the typical profile of the autofluorescence spectrum of the myoblasts, with a peak intensity at 515 nm. This spectrum profile is most probably the result of the convolution of different emission spectra related to several biological molecules inside the cells (see Monici (2005)). It is however very hard to deconvolve them.

Cellular pixels emit light whereas background pixels are associated with low light intensities. High intensity values for wavelengths having strong negative loadings (associated to cells in the first latent component) induce negative $\mathbf{T}_1$ scores, explaining why the cells in the $\mathbf{T}_1$ score images presented in Figure A-14 appear black on a white background. However positive loadings for the autofluorescence spectrum captured in the

second component induce high $\mathbf{T}_2$ scores, thus explaining why cellular pixels end-up more in the middle or the bottom-left corner in the $\mathbf{T}_1/\mathbf{T}_2$ score density histograms. Background pixels end-up in the top-right corner, as shown in Figure A-16.

With this technique, it appears that it is possible to segment easily in each image background pixels from cellular pixels. Unfortunately from an autofluorescence perspective myoblasts grown in SSM or SFM seem to have similar autofluorescence spectra. The main difference between the cells seems to be again their morphology (cells grown in SFM seem to be more elongated than cells grown in SSM). There are cells that emit more fluorescence than others, but that does not seem to be related to the culture medium itself. Further tests could be performed in the future to see if the degree of autofluorescence is related to cellular health or status (as suggested in the literature).

In conclusion, the autofluorescence hyperspectral microscopy technique would not necessarily help differentiating cells relative to the culture medium used. Autofluorescence generates a very weak signal compared to the signal generated by a fluorophore, and the experimental set-up might not be optimal to measure it with enough resolution inside the cells. However, this technique could be useful when using fluorophores (instead of relying on autofluorescence) that have emission spectra that are close. It would then be possible to measure the whole visible spectra in epifluorescence microscopy and to deconvolve the different spectra generated by several fluorophores.

A.6 Phase contrast hyperspectral microscopy with NIR camera (PCM-NIR)

For each experiment (January 6th, 11th and 16th 2013), PCM-NIR hyperspectral microscopy cubes associated with each well were analyzed through an overall PCA (one PCA per experiment, using a kernel MIA algorithm). Each slice image was divided by an average background (black) image (pre-processing step) in order to perform a correction for bad pixels. All the experiments analyzed gave similar results, so only the results associated to the experiment of January 6th 2013 are presented. The loadings, the score images and the score density histograms for the analysis of the experiment ran on January 6th 2013 are shown in Figure A-17, Figure A-18, Figure A-19, Figure A-20, Figure A-21, Figure A-22, Figure A-23 and Figure A-24.

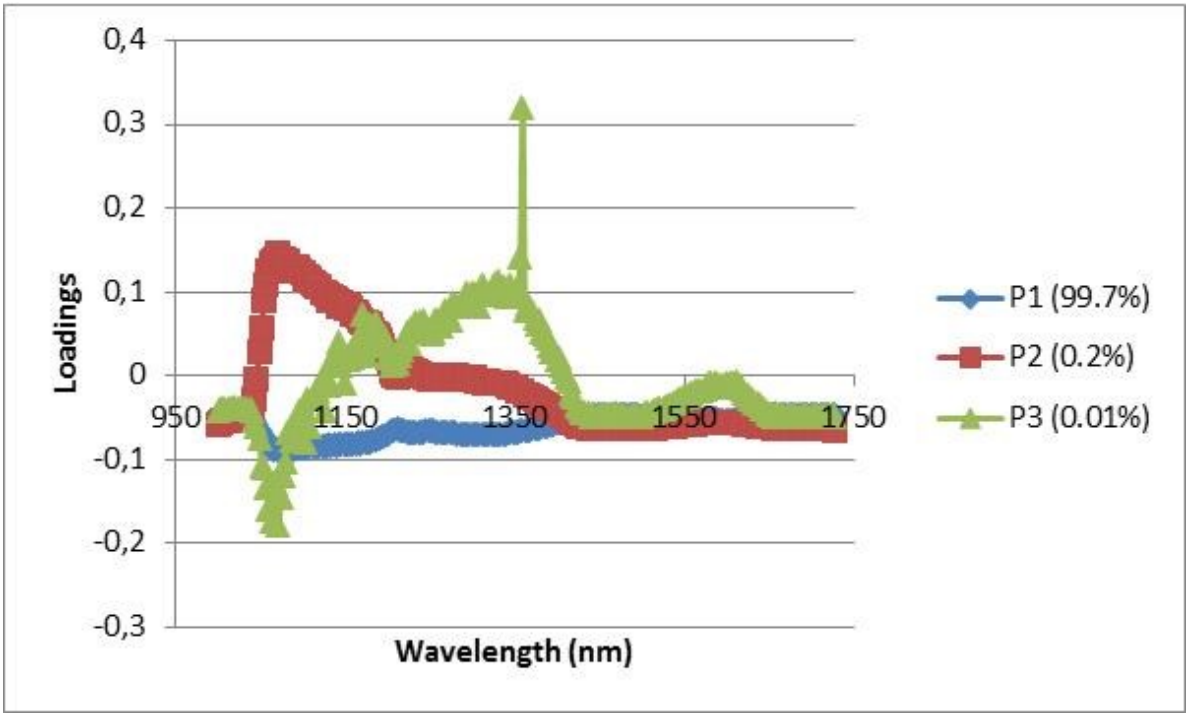


Figure A-17 : p_1, p_2 and p_3 loadings from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

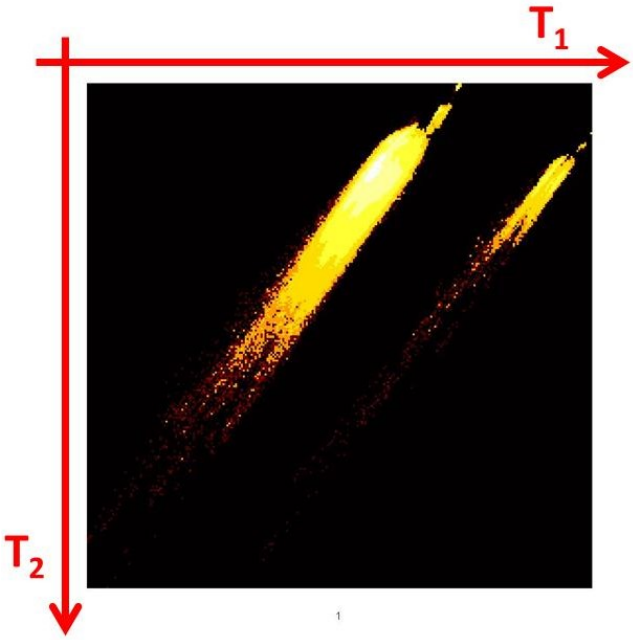


Figure A-18 : T_1/T_2 global score density histogram from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

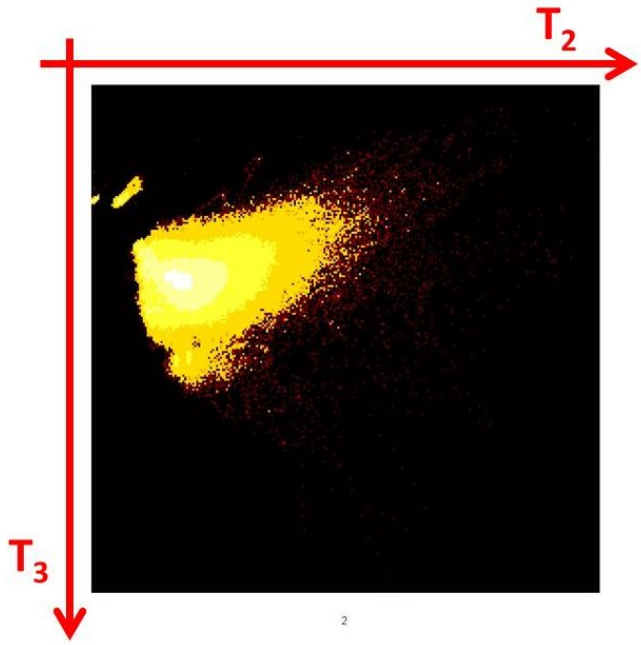


Figure A-19 :T_2/T_3 global score density histogram from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

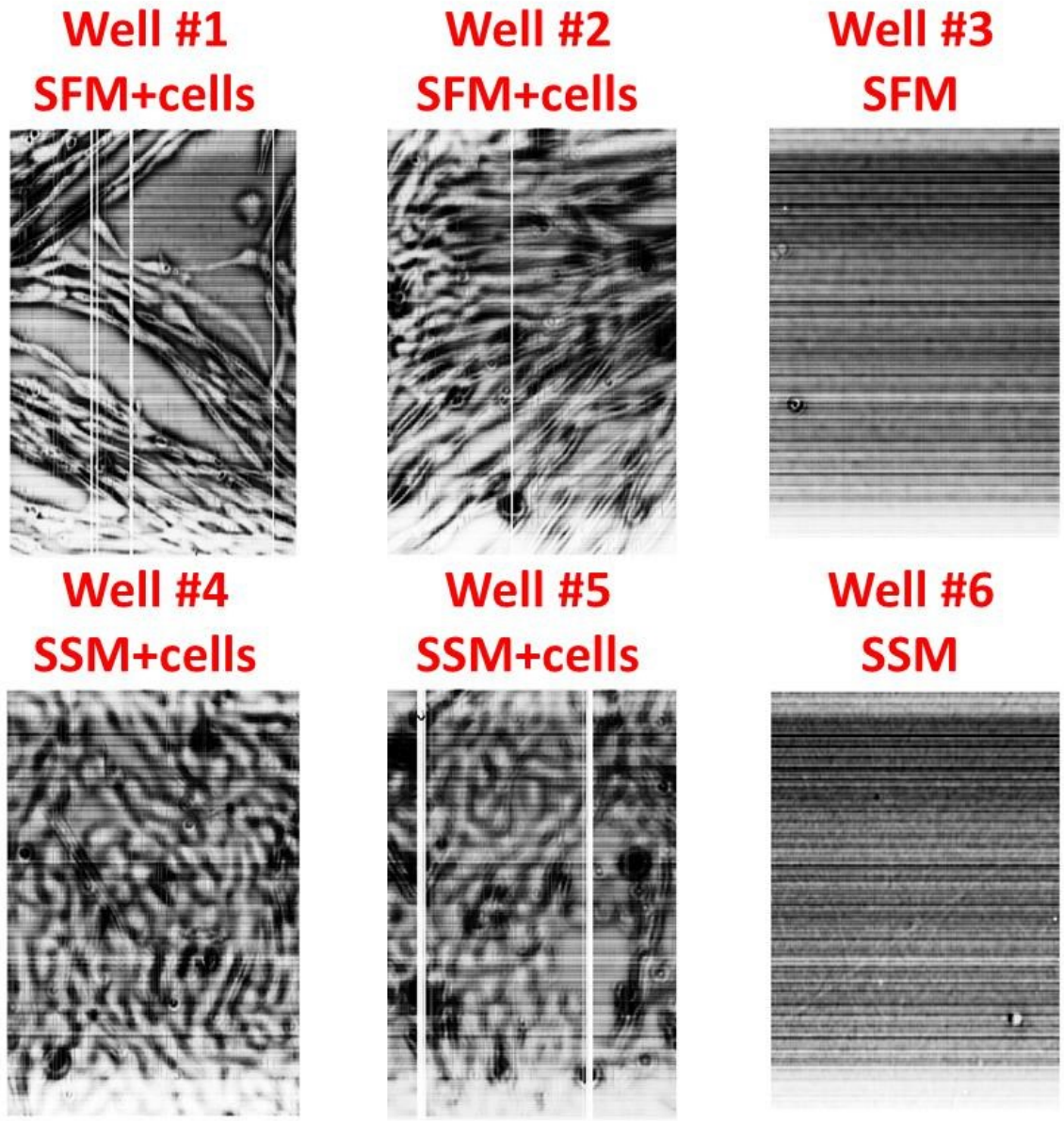


Figure A-20 : T_1 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

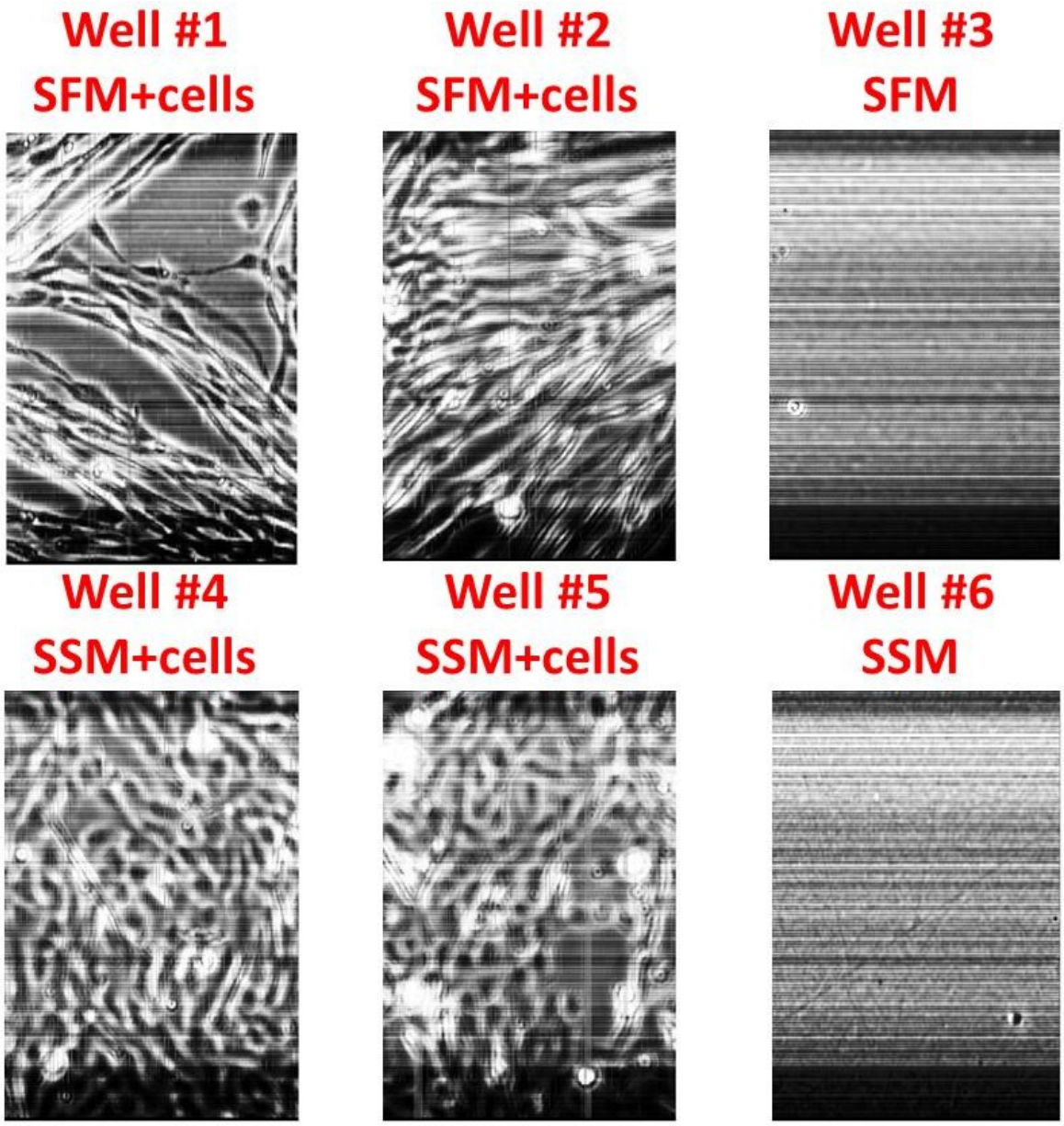


Figure A-21 : T_2 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

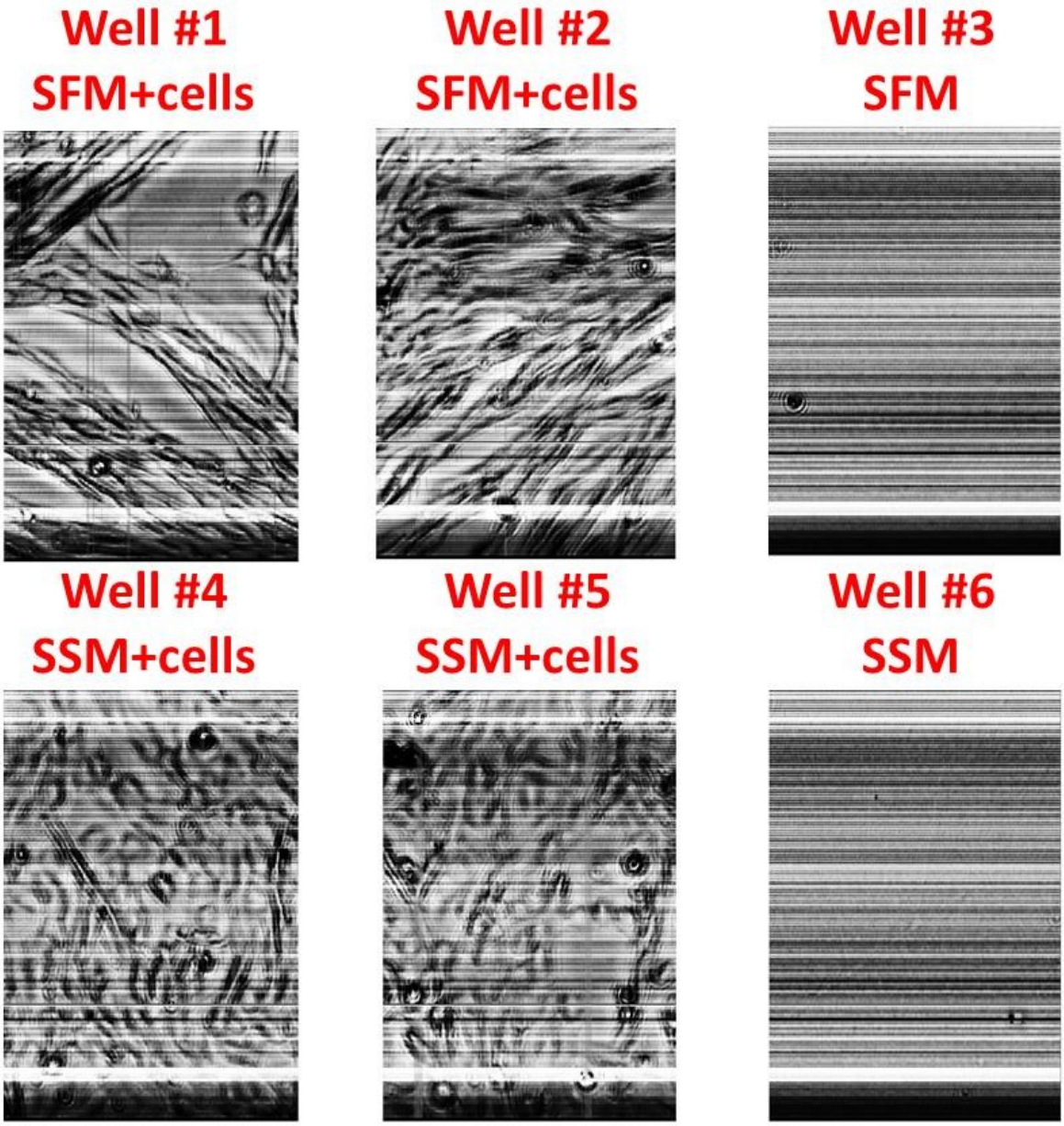


Figure A-22 : T_3 score images from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

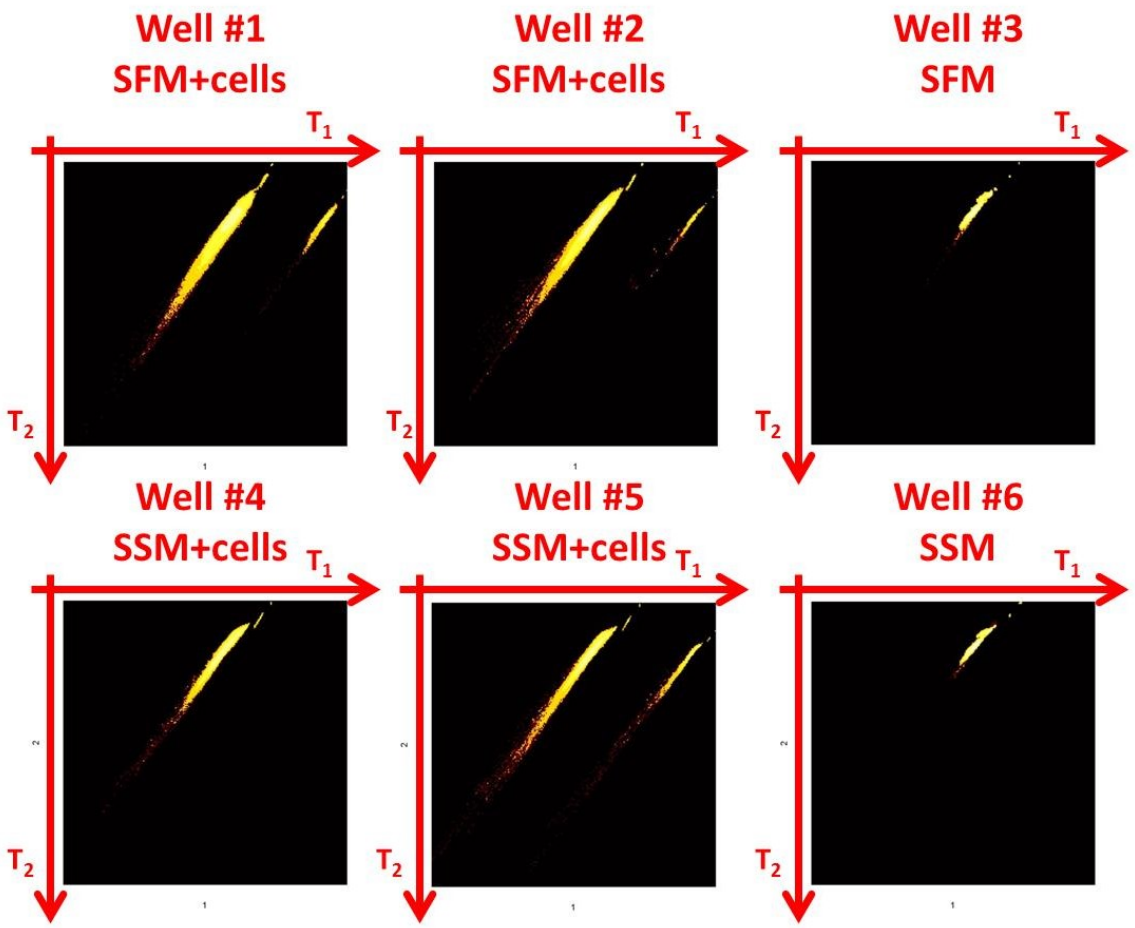


Figure A-23 : T_1/T_2 individual score density histograms from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

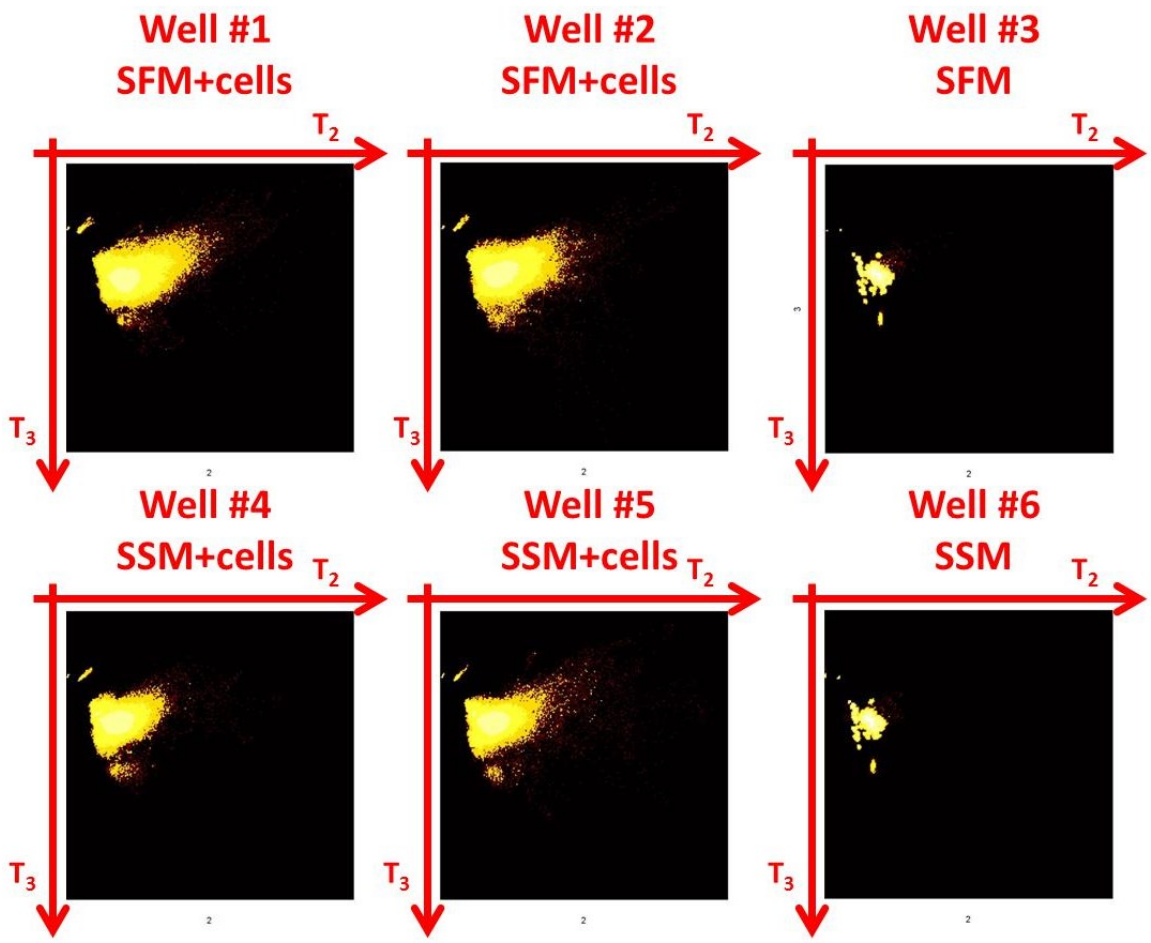


Figure A-24 : T_2/T_3 individual score density histograms from PCA performed on PCM-NIR hyperspectral cubes for experiment January 6th 2013

From a preliminary analysis of the loadings (Figure A-17), cellular membrane/border information would influence the 1110 to 1200 nm NIR spectrum (mainly captured in the second component) as cellular cytoplasm would influence the 1200 to 1420 nm NIR spectrum. The third component seems to distinguish between the cells cytoplasm and the cells membrane, however it is difficult to segment both cellular regions independently in the score density histograms. Masks and their corresponding overlays to segment background pixels and cellular pixels are presented in Figure A-25 and in Figure A-26 respectively.

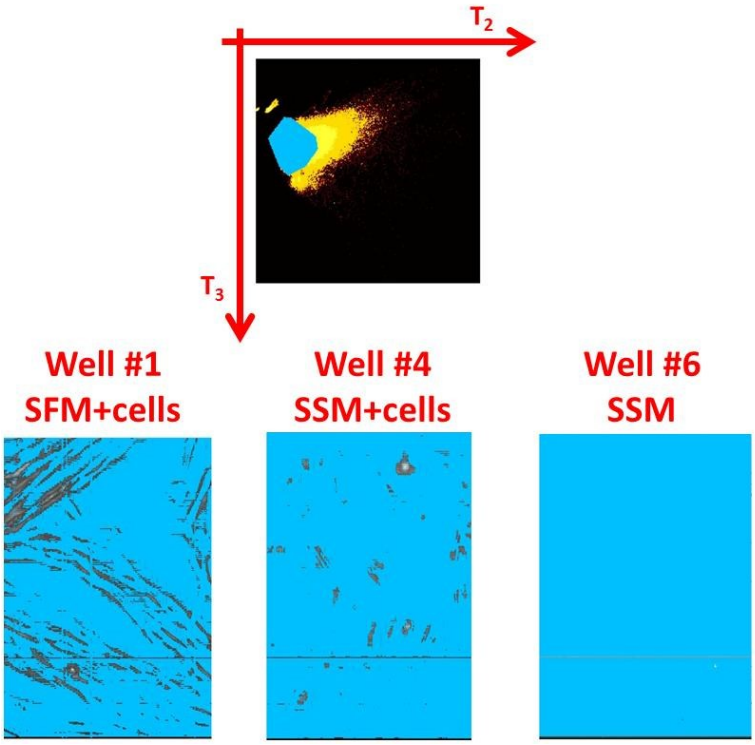


Figure A-25 : T_2/T_3 masking and overlays to segment background pixels (PCM-NIR hyperspectral cubes for experiment January 6th 2013)

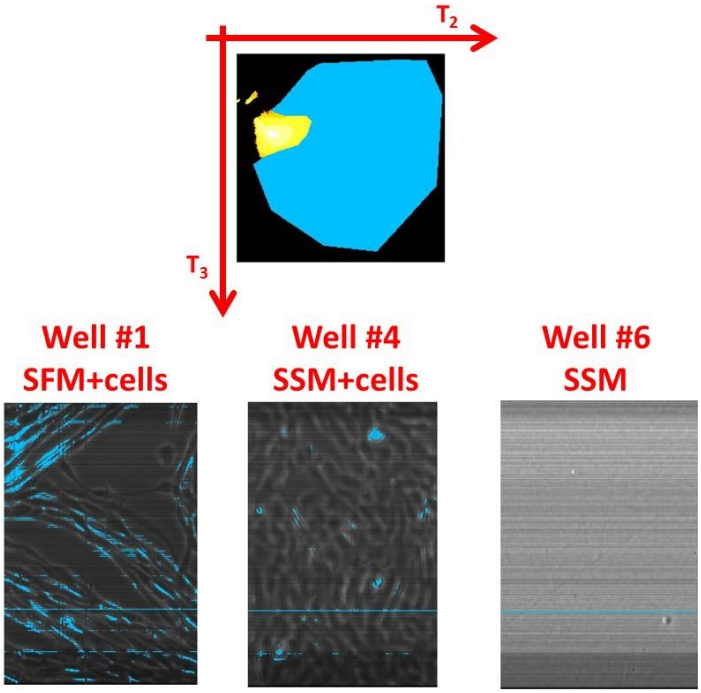


Figure A-26 : T_2/T_3 masking and overlays to segment cellular pixels (PCM-NIR hyperspectral cubes for experiment January 6th 2013)

There is not a lot of contrast between the cells cytoplasm and the background, and a sharp segmentation is difficult to achieve. This could be due to the fact that plastic multiwell plates were used during the tests (several chemical compounds in plastics can absorb light in the NIR range).

The first component is related with variations in light intensity transmitted through the sample in the NIR spectrum because all the loadings in this dimension have the same sign. In the NIR range, as the interior of the cells absorb light (medium $\mathbf{T}_1$ scores), cellular membranes that diffract light (bright halo around the cells in PCM) are associated with pixels with high intensities, explaining why cellular contours appear dark in the $\mathbf{T}_1$ score images (low $\mathbf{T}_1$ scores). One of the main difficulties was occasional NIR camera CCD problems, explaining why there are white vertical stripes in the $\mathbf{T}_1$ score images (appearing as a line at the extreme right of the $\mathbf{T}_1$/$\mathbf{T}_2$ score density histograms).

In conclusion, the PCM-NIR hyperspectral microscopy technique could help to segment cellular pixels from background pixels. There are several individual cells that seem to be brighter in score images (could be again an indication of cellular metabolism). The use of the NIR spectrum to assess the cellular metabolism could be explored in future work using for example specific fluorophores along epifluorescence microscopy. Despite the fact that there is no spectral global tendency to distinguish cells grown in SSM from cells grown in SFM, the information provided by the NIR spectrum could prove to be useful in future cell culture projects (maybe if the optical set-up is improved).

A.7 Conclusion

Whether using PCM-VIS, EM-VIS or PCM-NIR, there was no significant spectral difference detected between myoblasts grown in SSM and myoblasts grown in SFM. However in each case it was possible to see by eye the same difference in morphology (associated to textural information) in the score images: cells grown in SFM are more elongated and cells grown in SSM are rounder. These results explain partially why it was necessary to use the UWT-MIA method to distinguish the cells grown in the two culture media in a non-destructive manner (as described in Chapter 6).

Since all the tests performed so far were done using plastic 6 wells multiwell plates, there is a chance that the plastic could interfere with the measured spectra. Additional tests could be performed using glass petri dishes to see if a significant difference between cells grown in SSM and SFM could be detected.

Appendix B. Commercial products/software for microscopy, image acquisition and image processing

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
3D-DIAS	David R. Soll								X	X	Holographic method				X	X	Holographic method	(Heid, Voss et al. 2002; Hamahashi, Onami et al. 2005)
Acumen hci	TTP LabTech	X	CM				X											
Adobe Photoshop®	Adobe				X													(Miura 2005; Agley, Velloso et al. 2012)
BD Pathway 435	Beckton, Dickinson and Company	X	BF,CM	X	X													(Rimon and Schuldiner 2011)
BD Pathway 855	Beckton, Dickinson and Company	X	BF,CM	X	X													(Rimon and Schuldiner 2011)
Beckman Coulter IC 100	Beckman Coulter	X	BF	X														(Rimon and Schuldiner 2011)
BioImageXD	Universities of Jyväskylä and Turku in Finland, Max Planck Institute CBG in Dresden, Germany	X			X	X						X						(Eliceiri, Berthold et al. 2012)
Biostation-CT	Nikon	X	BF,PCM,EFM	X			X			X			X			X		
Braincells	Gabor Ivanesy					X					Manual	X					Manual	(Hand, Sun et al. 2009)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
Cedex XS System	Roche		BF	X	X		X			X								(Brinkmann, Lutkemeyer et al. 2002)
Celigo®	Brooks	X	BF,PCM,EFM	X			X			X								
Cell Observer or Cell Observer HS	Zeiss	X	BF,CM	X	X													(Rimon and Schuldiner 2011)
Cell Tracker	Bioanalytical Sciences Group, University of Manchester					X					Thresholding, Level Sets, Active Contours.	X					Edge Detection; Active Contours.	(Hand, Sun et al. 2009)
CellCognition	Gerlich laboratory at ETH Zurich	X					X				Local adaptive thresholding, watershed, split-and-merge.		X				SVM, Hidden Markov modelling.	(Held, Schmitz et al. 2010)
Cell^{M} / Cell^{R} (Cell^ series)	Olympus Europe		Not specified	X	X		X						X					(Rapoport, Becker et al. 2011)
CellMonitor™	Fraunhofer IPM		BF,PCM,EFM	X			X			X	Pattern recognition							
Cellomics ArrayScan VTI	Thermo Scientific	X	BF	X	X													(Rimon and Schuldiner 2011)
CellProfiler®	BROAD Institute	X			X		X			X	Several methods		X			X	Overlap, Distance, Feature Matching.	(Carpenter, Jones et al. 2006; Selinummi, Ruusuvuori et al. 2009; Quanli 2010)
CellScreen®	Innovatis AG, Germany		BF,PCM,EFM	X														(Brinkmann, Lutkemeyer et al. 2002)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
CellTrack	Database Research Group, The Ohio State University					X					Thresholding, Edge Detection.	X					Feature Matching; Mean-shifts; Optical Flow; Active Contours.	(Sacan, Ferhatosmanoglu et al. 2008; Hand, Sun et al. 2009)
CellViCAM	S. Burgemeister							X			Supervised machine learning							(Burgemeister, Nattkemper et al. 2010)
CyteSeer®	Vala Sciences	X					X											
Fiji					X													(Eliceiri, Berthold et al. 2012)
HCS A	Leica	X	BF,CM	X	X													(Rimon and Schuldiner 2011)
HoloMonitor®	Phase Holographic Imaging (PHI)		PCM,HL	X					X	X	Holographic, watershed.				X	X	Holographic	(Mõlder, Sebesta et al. 2008)
Icy®	Quantitative Image Analysis Unit at Institut Pasteur				X		X	X		X	Several plugins		X	X		X	Several plugins	
ImageJ®					X		X			X								(Polzer, Haasters et al. 2010)
MTrack2 (plugin)	Wayne Rasband						X				Thresholding, Edge Detection, Watersheds		X				Feature Matching	(Hand, Sun et al. 2009)
NIHimage	National Institutes of Health (NIH)				X		X			X								(Miura 2005)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
MosaicSuite: ParticleTracker 2D and 3D	MOSAIC Group, ETH Zurich				X								X			X	Several methods	
ImageMagick®	ImageMagick Studio LLC				X													(Satulovsky, Lui et al. 2008)
Image-Pro Plus®	Media Cybernetics			X	X	X						X						
ImageStreamX	Amnis	X	BF	X	X													(Rimon and Schuldiner 2011)
ImageXpress MICRO	Molecular Devices	X	BF	X														(Rimon and Schuldiner 2011)
ImageXpress Ultra	Molecular Devices	X	CM	X	X													(Rimon and Schuldiner 2011)
Imaris 5.5	Bitplane						X				Thresholding, Spot Detection.		X				Feature Matching.	(Hand, Sun et al. 2009)
IN Cell Analyzer 1000	GE Healthcare	X	BF,DIC,EFM	X	X													(Padfield, Rittscher et al. 2008; Padfield, Rittscher et al. 2011)
IN Cell Analyzer 2000	GE Healthcare	X	BF	X	X													(Rimon and Schuldiner 2011)
IN Cell Analyzer 3000	GE Healthcare	X	CM	X	X													(Rimon and Schuldiner 2011)
IncuCyte®	ESSEN BioScience	X	BF,PCM,EFM	X			X			X								
Insight Toolkit (ITK)	US National Library of Medicine of the National Institutes of Health				X		X			X	Filters							(Satulovsky, Lui et al. 2008)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
Juli™ / Juli™ *Br*	NanoEnTek		BF,EFM				X			X								
Large Scale Digital Cell Analysis System (LSDCAS)	Holden Comprehensive Cancer Center at the University of Iowa		PCM,EFM	X			X			X	Semi-automatic method (manual)							(Davis, Kosmacek et al. 2007)
MotoCell	Concita Cantarella et al.					X					Manual	X					Manual	(Cantarella, Sepe et al. 2009; Paduano, Sepe et al. 2010)
LabView®																		
CELLView	S. Gerardi		BF,PCM,EFM	X														(Gerardi, Galeazzi et al. 2005; Skoczylas, Rakowski et al. 2011)
Matlab®																		
CAFE MiCells	David W. Andrews, Mac Biophotonics Facility, McMaster University	X					X				Statistics, Neural network and bioinformatics toolboxes for classification.							
CellTracer®	Quanli Wang						X			X	Thresholding, Watershed		X			X	Proximity	(Quanli 2010)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
ZFIQ NeuronIQ NeuriteIQ DCellIQ GCellIQ AxonTracker AFINITI	Center for Bioengineering and Informatics, The Methodist Hospital System						X				Local adaptive thresholding, seeded watershed, fragment merging.		X				Feature Matching via Neighboring Graphs; Gradient Flow Tracking method.	(Liu, Nie et al. 2008; Sigoillot, Huckins et al. 2011)
Image Processing Toolbox®	The MathWorks Inc.				X													(Huth, Buchholz et al. 2010; Or-Tzadikario, Sopher et al. 2010; Al-Shanti, Faulkner et al. 2011; Rapoport, Becker et al. 2011; Topman, Sharabani-Yosef et al. 2011)
MOSAIC Group codes	MOSAIC Group, ETH Zurich						X			X	Deconvolution, snakes.		X			X	Feature point tracking.	
Time Lapse Analyzer®	Johannes Huth, Johann M. Kraus, Hans A. Kestler	X					X			X	Deconvolution, local thresholding, morphological operations.		X			X	Cell centroid detection, data association and state estimation via a Kalman filter.	(Huth, Buchholz et al. 2011)
Tri-track	Pascal Vallotton												X	X		X	Graph-based tracking	(Vallotton and Olivier 2013)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
Metamorph®	Molecular Devices			X	X													(Genovesio, Liedl et al. 2006; Ambuhl, Brepsant et al. 2012; Chao-Hui, Sankaran et al. 2012; Theriault, Walker et al. 2012)
Opera	PerkinElmer	X	CM,MPM	X	X		X				Features extraction							(Rimon and Schuldiner 2011)
Operetta	PerkinElmer	X	BF,CM	X	X													(Rimon and Schuldiner 2011)
QPM system	Pi Photonics, Inc.		HL	X					X		Holographic method							
Retrac	Molecular Motors Group					X					Manual	X					Manual	(Hand, Sun et al. 2009)
ScanImage	HHMI janelia farm research campus			X														
ScanR	Olympus	X	BF	X	X													(Rimon and Schuldiner 2011)
SigmaScan® Pro	Systat Software, Inc.				X													(Or-Tzadikario, Sopher et al. 2010)
Simi BioCell	Simi		DIC,HL						X	X	Holographic method				X	X	Holographic method	(Hamahashi, Onami et al. 2005)
TTT (Timm's Tracking Tool)	Timm Schroeder	X					X			X	Thresolding, features ,matching, semi-automatic.		X			X	Semi-automatic tracking	(Eilken, Nishikawa et al. 2009; Rieger, Hoppe et al. 2009)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
Vaa3D	Hanchuan Peng				X	X												(Eliceiri, Berthold et al. 2012)
Visual Basic (C/C++)					X		X			X	Region-growing, confidence-connected filter, level-set shape-detection filter		X			X	Region-growing, confidence-connected filter, level-set shape-detection filter	(Satulovsky, Lui et al. 2008)
OpenCV	OpenCV				X	X						X						(Eliceiri, Berthold et al. 2012)
OpenMPI library	Open MPI Development Team, Indiana University.				X		X			X	Wavelets, SVM							(Skoczylas, Rakowski et al. 2011)
Particle Tracking	MOSAIC Group, ETH Zurich												X				Feature point tracking.	
ScaLAPACK library	Univ. of Tennessee; Univ. of California, Berkeley; Univ. of Colorado Denver; and NAG Ltd.				X		X			X	Wavelets, SVM							(Skoczylas, Rakowski et al. 2011)
Visualization Toolkit (VTK)	Will Schroeder, Ken Martin, and Bill Lorensen				X													(Eliceiri, Berthold et al. 2012)

Product Name	Company / Author	HTS	Microscopy Optics for Image Acquisition	IA	Image Processing													References
						Cells Segmentation						Cells Tracking						
					MPS	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	US	EFM CM MPM	DF	HL	BF PCM DIC	Method used	
Volocity 4	Improvision						X				Thresholding, Edge Detection, Watersheds		X				Feature Matching	(Hand, Sun et al. 2009)
WiScan	IDEA-biomedicals	X	BF	X	X													(Rimon and Schuldiner 2011)
µManager	Vale Lab			X														

BF: Brightfield

CM: Confocal Microscopy

DF: Darkfield

DIC: Differential Interference Contrast

EFM: Epifluorescence Microscopy

HL: Holography

HTS: High-Throughput Screening

IA: Image Acquisition

MPM: Multiphotons Microscopy

MPS: Multipurpose Software

PCM: Phase Contrast Microscopy

US: Unspecified

Appendix C. Protocols used for the culture of human myoblasts

Most of the laboratory protocols used for the culture of human myoblasts were inspired and adapted from the protocols established by a former graduate student (Mr. Victor Alain Parent), which were used for the LOBSFM medium project (Parent 2009).

C.1 Frozen cells thawing and seeding

- Pull out a human myoblasts vial from the liquid nitrogen tank (myoblasts from Dr. Tremblay's lab, BB13M P3, P4 or P5);
- Heat the vial slowly by storing it in a warm incubator at 37 ^{0}C for about 5 minutes (until a small portion of the ice melted);
- Under a laminar flow hood, slowly add culture medium at 37 ^{0}C in the vial and mix smoothly using a pipette until there is no ice left;
- Dilute in 3-4 mL of culture medium and perform a centrifugation at 1500 RPM for 5 minutes;
- Remove the supernatant (to remove the DMSO) and resuspend the cells in fresh culture medium (3-4 mL) to create a mother solution;
- Perform a hemacytometer count on the mother solution;
- Seed a T-flask (25 cm^2) full of fresh culture medium (5 mL) with the mother solution (0.5 mL, targeting between 2,000 to 50,000 cells/mL): this T-flask is used to expand cells for the next passage;
    - Before putting the T-flask in the incubator, agitate it well to have a uniform distribution of the cells using a cross pattern while keeping it on a steady horizontal surface (up-down, left-right), avoiding circular patterns to prevent cells from accumulating at the center;
    - Unscrew slightly the T-flask cap to favor gas diffusion;
- Dilute the mother solution in a 15 mL plastic tube to reach an appropriate cell density (usually 0.5 mL of mother solution for 11.5 mL of fresh medium) to create a daughter solution;
- Agitate the plastic tube vigorously to make sure that the cells are well spread into the liquid volume (use a shaker);
- Fill each well of a 6-wells multiwell plate with 2 mL of daughter solution;
    - Agitate the multiwell plate to obtain a uniform distribution of the cells using a cross pattern while keeping it on a steady horizontal surface (up-

down, left-right), avoiding circular patterns to prevent cells from accumulating at the center of each well;

- The 6-wells multiwell plate is used for live-cell imaging;
- When the cells in the T-flask reach 80% confluence, trypsinize the culture to harvest the cells (passaging);
- Keep the cultures at 37.5 ^{0}C and at 5% CO_2: the culture medium should be changed under a laminar flow hood every 48-72 hours.

C.2 Cell culture trypsinization and passaging

- Under a laminar flow hood, open the T-flask (25 cm^2) and remove the old culture medium;
- Wash the cells with a PBS solution (around 5 mL) to remove as much proteins as possible;
- Remove the spent PBS solution;
- Add around 1 mL of trypsin (TrypsinLE);
- Leave at ambient temperature for 5-10 minutes until the cells detach (usually it is longer for SSM than for SFM);
    - Strike vigorously the T-flask on a flat surface to help the cells to detach (be careful not to brake the T-flask);
    - Meanwhile, fill a plastic tube with 4 mL of fresh SSM or with 3 mL of SFM and 1 mL of a solution of trypsin inhibitor (soy bean proteins): this will inhibit the trypsin used;
- Once the cells are detached, remove the solution from the T-flask and pipette it into the plastic tube to inhibit the trypsin;
- Agitate the plastic tube (use a shaker);
- Perform a centrifugation at 1500 RPM for 5 minutes;
- Remove the supernatant and resuspend the cells in fresh culture medium (3-4 mL) to create a mother solution;
- Perform a hemacytometer count on the mother solution;
- Seed a T-flask (25 cm^2) full of fresh culture medium (5 mL) with the mother solution (0.5 mL, targeting between 2,000 to 50,000 cells/mL): this T-flask will expand cells for the next passage;
    - Before putting the T-flask in the incubator, agitate it well to have a uniform distribution of the cells using a cross pattern while keeping it on a steady

horizontal surface (up-down, left-right), avoiding circular patterns to prevent cells from accumulating at the center;
    - Unscrew the T-flask cap slightly to favor gas diffusion;
- Dilute the mother solution in a 15 mL plastic tube to reach an appropriate cell density (usually 0.5 mL of mother solution for 11.5 mL of fresh medium) to create a daughter solution;
- Agitate the plastic tube vigorously to make sure that the cells are well spread into the liquid volume (use a shaker);
- Fill each well of a 6-wells multiwell plate with 2 mL of daughter solution;
    - Agitate the well plate to have a uniform distribution of the cells using a cross pattern while keeping it on a steady horizontal surface (up-down, left-right), avoiding circular patterns to prevent cells from accumulating at the center of each well;
- The 6-wells multiwell plate is used for live-cell imaging;
- When the cells in the T-flask reach 80% confluence, trypsinize the culture to harvest the cells (passaging);
- Keep the cultures at 37.5 ^{0}C and at 5% CO_2: the culture medium should be changed under a laminar flow hood every 48-72 hours.

C.3 Cells freezing protocol

- Trypsinize the cells from a T-flask (as described previously) to produce in a plastic tube a mother solution having a cellular density of 1-3 x 10^6 cells/mL (in SSM medium);
- Add DMSO to obtain a concentration of 10% DMSO (cellular suspension);
- Mix well (using a shaker) and put on ice;
- Fill small 1.8 mL cryogenic vials with 1 mL of the cellular suspension (SSM + 10% DMSO);
- Place the vials in a Styrofoam box full of ice and leave overnight in a -80 ^{0}C freezer (the temperature inside the vials will decrease of 1 ^{0}C per minute for 12-15 hours);
- Store the vials in liquid nitrogen.

C.4 Hemacytometer cell count

- Recover a sample from a cellular mother solution;
- In an Eppendorf, mix a specified volume of the sample with erythrosine B depending of the cell concentration in the mother solution (start with 100 µL of mother solution and 100 µL of erythrosine B);
    - Dilution with erythrosine B (1 mg/mL) should be done in order to have at least a mean cell count of 15 cells per hemacytometer section;
    - Mix well with a pipette;
- Use a pipette to send 10 µL of the mix in each hemacytometer chamber;
- Place the hemacytometer under a microscope and count colored cells (dead cells) and white/transparent cells (living cells);
    - Cell counts should be performed in the first two minutes following the dilution, since erythrosine B will start penetrating living cells as well after this delay;
- Count the number of living cells in each hemacytometer quadrant and sum them (total number of cells counted);
- Divide the total number of cells counted by the number of hemacytometer quadrants and multiply by the erythrosine B dilution factor and by the appropriate hemacytometer chamber factor (here 10,000);
- The computation is summarized by the following equation:

$$\frac{Cells}{mL} = \left(\frac{Total\ number\ of\ cells\ counted}{Number\ of\ quadrants}\right) * (Erythrosine\ B\ dilution) * (10{,}000) \qquad \text{(C-1)}$$

C.5 Preparation of serum-supplemented medium (SSM)

Table C-1 provides a list of the different ingredients required to prepare 200 mL of a serum-supplemented medium (SSM) (with the required quantities):

Table C-1 : Serum-supplemented medium (SSM) ingredients

Compounds	Concentration in culture medium	Units	Aliquot concentration (µg/mL)	Liquid volume per aliquot (µL)	Quantities required	Units
MB-I	85%	%	-	-	170	mL
FBS	15%	%	-	-	30	mL
Dexametazone	0,39	ug/mL	20	1000	4	Aliquots
ITS	1	X	-	-	2	mL
bFGF	10	ng/mL	80	25	1	Aliquot
BSA	0,5	mg/mL	-	-	100	mg

Steps:

- Sterilize a 500 mL glass bottle and required pipette tips;
- Under a laminar flow hood, fill the glass bottle with the required amount of MB-I;
- Weigh the required amount of BSA in a hexagonal weigh plastic dish;
- Under the laminar flow hood, add with the pipette gun (attached to a disposable pipette) the FBS, the dexametazone and the ITS;
- Mix by pushing/pulling a couple of times using the pipette gun to dissolve the BSA;
- Add with a pipette (and the appropriate pipette tip) the bFGF to the liquid;
- With a 20 mL syringe, filter the content of the plastic dish above the glass bottle with a 0.45 µm disposable filter (sterile filtration). NOTE: depending on the FBS lot used, it may be required to use two or three disposable filters;
- Close the glass bottle and mix well;
- Store the culture medium at 4 ^{0}C.

C.6 Preparation of serum-free medium (SFM)

Table C-2 provides a list of the different ingredients required to prepare 200 mL of a serum-free medium (SFM) (with the required quantities):

Table C-2 : Serum-free medium (SFM) ingredients

Compounds	Concentration in culture medium	Units	Aliquot concentration (µg/mL)	Liquid volume per aliquot (µL)	Quantities required	Units
MB-I	33%	%	-	-	66,7	mL
F12	33%	%	-	-	66,7	mL
RPMI	33%	%	-	-	66,7	mL
ITS	1	X	100	-	2	mL
Chemically Defined Lipid Concentrate	1	X	100	-	2	mL
B27-VitA	1	X	50	1000	4	Aliquots
FGF 2	10	ng/mL	80	25	1	Aliquot
FGF 4	10	ng/mL	80	25	1	Aliquot
IGF-1	10	ng/mL	50	100	0,4	Aliquot
EGF	10	ng/mL	80	25	1	Aliquot
IL-1alpha	5	ng/mL	25	40	1	Aliquot
IL-1beta	5	ng/mL	25	40	1	Aliquot
Dexametazone	0,39	ug/mL	20	1000	4	Aliquots
BSA	0,5	mg/mL	-	-	100	mg
Fetuin	0,5	mg/mL	-	-	100	mg
Glutamin	4	mM	200	-	4	mL
Fibronectin	5	ug/mL	1000	-	1	mL

Steps:

- Sterilize a 500 mL glass bottle and required pipette tips;
    - The night before, coat the glass bottle with Sigmacote® and leave it to dry overnight (since it is a serum-free medium, this prevents cytokines from sticking to the glass);
- Under a laminar flow hood, fill the glass bottle with the required amount of MB-I, F12 and RPMI;
- Weigh the required amounts of BSA and of fetuin in a hexagonal weigh plastic dish;
- Under the laminar flow hood, add with the pipette gun (attached to a disposable pipette) the dexametazone, the B27, the ITS, the lipids mix and the glutamin;
- Mix by pushing/pulling a couple of times using the pipette gun to dissolve the solids;

- Centrifuge at 10,000 RPM for 1 minute the growth factors aliquots;
- Add with a pipette (and the appropriate pipette tip) the growth factors to the liquid: FGF-2, FGF-4, IGF-1, EGF, IL-1alpha and IL-1beta;
- Mix by rotating slightly the plastic dish on the flat surface under the laminar flow hood;
- With a 20 mL syringe, filter the content of the plastic dish above the glass bottle with a 0.45 µm disposable filter (sterile filtration);
- With a pipette, add directly the fibronectin into the glass bottle;
- Close the glass bottle and mix well;
- Store the culture medium at 4 ^{0}C.

C.7 Compounds for cell culture

Table C-3 provides a list of the compounds that were used for cell culture and their supplier:

Table C-3 : List of compounds used for cell culture and their suppliers

Compound	Supplier	Product ID	Quantity	Concentration
MB-I	MCDB120 from Dr. Jacques Tremblay's Lab (CRCHUL)	-	1000 mL	-
F12	Life Technologies/Gibco	11765-054	500 mL	-
RPMI	Life Technologies/Gibco	21870-076	500 mL	-
FBS	HyClone / Thermo Fisher	SH3039603	500 mL	-
ITS	Life Technologies/Gibco	41400-045	10 mL	100X
Chemically Defined Lipid Concentrate	Life Technologies/Gibco	11905-031	100 mL	-
B27-VitA	Life Technologies/Gibco	12587-010	10 mL	50X
FGF 2	Feldan Bio	1D-07-017b	50 µg	-
FGF 4	Feldan Bio	1D-07-001b	25 µg	-
IGF-1	Feldan Bio	1D-16-002b	100 µg	-
EGF	Feldan Bio	1D-06-005b	500 µg	-
IL-1alpha	Feldan Bio	1D-11-004b	10 µg	-
IL-1beta	Feldan Bio	1D-11-042b	10 µg	-
Dexametazone	Sigma-Aldrich	D8893-1MG	1 mg	-
BSA	Sigma-Aldrich	A1933-25G	25 g	-
Fetuin	Sigma-Aldrich	F2379-1G	1 g	-
Glutamin	Life Technologies/Gibco	25030-081	100 mL	200 mM
Fibronectin	Millipore	FC010-10MG	10 X 1 mL	1 mg/mL
Trypsin inhibitor from Glycine max (soybean)	Sigma-Aldrich	T6522-25MG	25 mg	-
Trypsin LE	Life Technologies/Gibco	12604-021	500 mL	1X
PBS	Life Technologies/Gibco	10010-023	500 mL	1X

- Growth factors were aliquoted to the required concentration using ultrapure water;
- The newly purchased dexamethazone powder (inside a brown glass bottle) was dissolved with 1 mL absolute ethanol in the bottle (anhydrous) and 49 mL F12 medium before being aliquoted in 50 aliquots of 1 mL (aliquot concentration of 20 µg/mL);
- The trypsin-inhibitor powder was dissolved in 10 mL PBS before being filtered at 0.45 µm in a plastic tube and stored at 4 ^{0}C.

C.8 Composition of MCDB120

The composition of the MCDB120 medium is provided in Table C-4 according to Ham and Blau (1988).

Table C-4 : Composition of MCDB120 medium (Ham and Blau 1988)

	M/L	mg/L		M/L	mg/L
Amino Acids			**Other Organic Components**		
L-Alanine	3.00E-5	2.67	Adenine	1.00E-6	0.1351
L-Arginine.HCl	1.00E-3	210.67	Choline Chloride	1.00E-4	13.96
L-Asparagine.H_20	1.00E-4	15.01	D-Glucose	5.55E-3	1000.00
L-Aspartic Acid	1.00E-4	13.31	myo-Inositol	1.00E-4	18.016
L-Cysteine.HCl.H_20	2.00E-4	35.13	Putrescine.2HCl	1.00E-9	0.0001611
L-Glutamic Acid	3.00E-5	4.41	Sodium Pyruvate	1.00E-3	110.04
L-Glutamine	1.00E-2	1461.50	Thymidine	1.00E-7	0.02422
Glycine	3.00E-5	2.25	**Bulk Inorganic Salts**		
L-Histidine.HCl.H_20	2.00E-4	41.93	$CaCl_2.2H_20$	1.60E-3	235.23
L-Isoleucine	5.00E-4	65.58	KCl	4.00E-3	298.20
L-Leucine	1.00E-3	131.17	$MgSO_4.7H_20$	1.00E-3	246.38
L-Lysine.HCl	1.00E-3	181.65	NaCl	1.10E-1	6430.0
L-Methionine	2.00E-4	29.84	$Na_2HPO_4.7H_2O$	5.00E-4	134.04
L-Phenylalanine	2.00E-4	33.04	**Trace Elements**		
L-Proline	1.00E-4	11.51	$CuSO_4.5H_20$	1.00E-8	0.002496
L-Serine	3.00E-4	31.53	$FeSO_4.7H_20$	3.00E-6	0.8340
L-Threonine	3.00E-4	35.73	H_2SeO_3	3.00E-8	0.00387
L-Tryptophan	2.00E-5	4.08	$MnSO_4.5H_20$	1.00E-9	0.000241
L-Tyrosine	1.00E-4	18.12	$Na_2SiO_3.9H_20$	1.00E-5	2.842
L-Valine	1.00E-3	117.15	$(NH_4)_6Mo_7O_{24}.4H_20$	3.00E-9	0.00371
Vitamins			NH_4VO_3	5.00E-9	0.000585
d-Biotin	3.00E-8	0.00733	$NiCl_2.6H_20$	3.00E-10	0.0000713
Folinic Acid (Ca salt).$5H_20$	1.00E-6	0.602	$ZnSO_4.7H_20$	3.00E-7	0.08625
DL-Alpha-Lipoic Acid	1.00E-8	0.002063	**Buffers, Indicators and Miscellaneous**		
Niacinamide	5.00E-5	6.11	Phenol-Red (Na salt)	3.30E-6	1.242
D-Pantothenic Acid (Hemi-Ca salt)	1.00E-4	23.82	$NaHCO_3$	1.40E-2	1176.0
Pyridoxine.HCl	1.00E-5	2.056			
Riboflavin	1.00E-8	0.003764			
Thiamin.HCl	1.00E-5	3.373			
Vitamin B12	1.00E-8	0.01355			

Appendix D. List of PCM image datasets generated during the project

Experiment Name	Start Date	Cells	Medium	Cell initial concentration (c/mL)	Comments
Experiment_January_25_2011	January 25, 2011	BB13MP7 (myoblasts)	SFM	42500	Done with fresh B27 vitamin (added 4mL instead of 2mL). Excellent results!
Experiment_January_31_2011	January 31, 2011	BB13MP8 (myoblasts)	SFM	43125	Same thing as previous experiment
Experiment_February_07_2011	February 7, 2011	BB13MP9 (myoblasts)	SFM	39375	With fresh Fibronectine
Experiment_February_14_2011	February 14, 2011	BB13MP10 (myoblasts)	SFM	26875	With fresh Fibronectine
Experiment_April_08_2011	April 8, 2011	BB13MP5 (myoblasts, VP)	SSM (1 to 3) and SFM (4 to 6)	8750	New cell batch. This time I used the FBS that was in aliquots from Victor. Cells seem apoptotic.
Experiment_April_21_2011	April 21, 2011	BB13MP6 (myoblasts, VP)	SSM (1 to 3) and SFM (4 to 6)	1458 c/mL for SSM, 47500 c/mL for SFM	Filters plugged, and no CO2 was going to the multiwell plate on Day 5. Apoptotic cells.
Experiment_April_26_2011	April 21, 2011	BB13MP6 (myoblasts, VP) TÉMOIN	SSM (1 to 3) and SFM (4 to 6)	1458 c/mL for SSM, 47500 c/mL for SFM	Reference plate from previous experiment (April 21st 2011). The cells started to grow well when I switched to the old FBS from Victor. The cells were apoptotic before.
Experiment_May_08_2011	April 28, 2011	BB13MP7 (myoblasts, VP)	SFM only	75000	Only one image acquisition
Experiment_May_14_2011	May 6, 2011	BB13MP8 (myoblasts, VP)	SFM only	41250	Only one image acquisition
Experiment_May_21_2011	May 13, 2011	BB13MP9 (myoblasts, VP)	SFM only	39792	Only one image acquisition
Experiment_May_22_2011	May 22, 2011	BB13MP7 SSM et BB13MP10 SFM (myoblasts, VP)	SSM (1 to 3) and SFM (4 to 6)	23542 for SSM and 47917 for SFM	Use of Victor FBS. Cells were fusing in SFM.
Experiment_May_27_2011	May 26, 2011	BB13MP8 SSM et BB13MP11 SFM (myoblasts, VP)	SSM (1 to 3) and SFM (4 to 6)	51771 for SSM and 26354 for SFM	Use of Victor FBS. No CO2 at the end…
Experiment_June_01_2011	May 26, 2011	BB13MP8 SSM et BB13MP11 SFM (myoblasts, VP) TÉMOIN	SSM (1 to 3) and SFM (4 to 6)	51771 for SSM and 26354 for SFM	Use of Victor FBS. Reference plate.

Experiment Name	Start Date	Cells	Medium	Cell initial concentration (c/mL)	Comments
Experiment_June_17_2011	June 17, 2011	BB13MP11 SSM et BB13MP14 SFM (myoblasts, VP)	SSM (1 to 3) and SFM (4 to 6)	28646 for SSM and 7292 for SFM	Use of Corinne FBS (fresh FBS, new medium batch).
Experiment_June_20_2011	June 17, 2011	BB13MP11 SSM et BB13MP14 SFM (myoblasts, VP) TÉMOIN	SSM (1 to 3) and SFM (4 to 6)	28646 for SSM and 7292 for SFM	Use of Corinne FBS (fresh FBS, new medium batch).
Experiment_June_25_2011	June 24, 2011	BB13MP12	SSM only	19896	Use of Corinne FBS (fresh FBS, new medium batch).
Experiment_July_21_2011	July 21, 2011	BB13MP16	SSM only	9688	Use of Corinne FBS (fresh FBS, new medium batch).
Experiment_August_03_2011	August 3, 2011	BB13MP17 (myoblasts, VP)	SSM	4688	
Experiment_August_11_2011	August 10, 2011	BB13MP18 (myoblasts, VP) in wells 1 to 3 and BB13MP5 (from vials) in wells 4 to 6	SSM	1458 for BB13MP18 and 10000 for BB13MP5	
Experiment_August_19_2011	August 16, 2011	BB13MP6 (myoblasts)	SSM	30417	
Experiment_August_22_2011	August 21, 2011	BB13MP7 (myoblasts)	SSM	12500	
Experiment_August_26_2011	August 26, 2011	BB13MP8 (myoblasts)	SSM	24271	
Experiment_August_31_2011	August 31, 2011	BB13MP9	SSM only	18542	
Experiment_September_06_2011	September 5, 2011	BB13MP10	SSM only	11667	
Experiment_September_11_2011	September 11, 2011	BB13MP11	SSM only	14688	
Experiment_September_18_2011	September 17, 2011	BB13MP12	SSM only	10000	
Experiment_September_24_2011	September 23, 2011	BB13MP13	SSM only	10313	
Experiment_October_01_2011	September 30, 2011	BB13MP14	SSM only	7813	
Experiment_October_08_2011	October 8, 2011	BB13MP15	SSM only	7500	
Experiment_October_17_2011	October 16, 2011	BB13MP16	SSM only	4479	
Experiment_November_04_2011	October 27, 2011	BB13MP17	SSM only	625	
Experiment_November_08_2011	November 8, 2011	BB13MP6	SFM only	14167	
Experiment_November_14_2011	November 13, 2011	BB13MP7	SFM only	19271	
Experiment_November_18_2011	November 17, 2011	BB13MP8	SFM only	18542	

Experiment Name	Start Date	Cells	Medium	Cell initial concentration (c/mL)	Comments
Experiment_November_23_2011	November 22, 2011	BB13MP9	SFM only	22188	
Experiment_November_28_2011	November 27, 2011	BB13MP10	SFM only	21458	
Experiment_December_03_2011	December 2, 2011	BB13MP11	SFM only	11458	
Experiment_December_09_2011	December 8, 2011	BB13MP12	SFM only	5833	
Experiment_December_16_2011	December 15, 2011	BB13MP13	SFM only	7083	
Experiment_December_23_2011	December 22, 2011	BB13MP14	SFM only	7708	
Experiment_January_16_2012	January 16, 2012	BB13MP6	SSM (1 to 3) and SFM (4 to 6)	4792	Differenciation experiment (wells 5 and 6). Cells in SSM (wells 1,2,3) are in senescence (SSM prepared in november 2011)
Experiment_January_27_2012	January 26, 2012	BB13MP5	SSM (1 to 3) and SFM (4 to 6)	17500	Differenciation experiment (wells 5 and 6). Cells in SSM (wells 1,2,3) are in senescence (SSM prepared in november 2011)
Experiment_February_04_2012	February 3, 2012	BB13MP4	SSM (1 to 3) and SFM (4 to 6)	31250	Differenciation experiment (wells 5 and 6). Cells in SSM (wells 1,2,3) are in senescence (SSM prepared on February 2nd 2012). Differenciation medium in wells 2 and 3 on Trial 8.
Experiment_February_13_2012	February 13, 2012	BB13MP5	SSM (1 to 3) and SFM (4 to 6)	8750	Since we have problems with SSM, I am using the old SSM medium batch from 01/11/2011 (seems to work well)
Experiment_February_26_2012	February 25, 2012	BB13MP8	SSM (1 to 3) and SFM (4 to 6)	14167	Cells have a very low survivability… NO GROWTH
Experiment_February_27_2012	February 27, 2012	BB13MP7	SSM (1 to 3) and SFM (4 to 6)	5000	Use the SFM batch from 13/12/2011 and SSM medium from 01/11/2011
Experiment_March_10_2012	March 10, 2012	BB13MP9	SSM, with different concentrations of ITS	11771	
Experiment_March_17_2012	March 16, 2012	BB13MP10	SSM, with different concentrations of ITS	7656	Just before the trypsinization of the previous passage, the cells were starting to fuse

Experiment Name	Start Date	Cells	Medium	Cell initial concentration (c/mL)	Comments
Experiment_March_22_2012	March 21, 2012	BB13MP11	SSM, with different concentrations of ITS	6927	
Experiment_March_29_2012	March 29, 2012	BB13MP5	SFM, with different concentrations of B27	1667	
Experiment_April_20_2012	April 19, 2012	BB13MP9	SFM, with different concentrations of B27	3594	CAUTION: I made the mother solution (for dilutions) with the FBS medium: this means that the cells grew in the last couple of days in SFM supplemented with some SSM… so this is why there is not a lot of difference between wells
Experiment_April_24_2012	April 24, 2012	BB13MP10	SFM, with different concentrations of B27	4948	
Experiment_May_03_2012	May 3, 2012	BB13MP6	SFM, with different concentrations of B27	2813	
Experiment_May_09_2012	May 8, 2012	BB13MP7	SFM, with different concentrations of B27	14479	
Experiment_May_13_2012	May 12, 2012	BB13MP8	SSM, with different concentrations of FBS	8646	
Experiment_June_02_2012	June 1, 2012	BB13MP12	SSM, with different concentrations of FBS	2448	
Experiment_June_18_2012	June 18, 2012	BB13MP7	SSM, with different concentrations of FBS	10938	

Experiment Name	Start Date	Cells	Medium	Cell initial concentration (c/mL)	Comments
Experiment_June_24_2012	June 23, 2012	BB13MP8	SSM, with different concentrations of FBS	10573	

Appendix E. Metamorph® image acquisition procedure (IX81 Olympus Microscope)

To acquire images from the IX-81 Olympus™ microscope, a series of steps are required to ensure an optimal live-cell imaging set-up. The software used for image acquisition is Metamorph® (Molecular Devices™). The Metamorph® general interface is shown in Figure E-1.

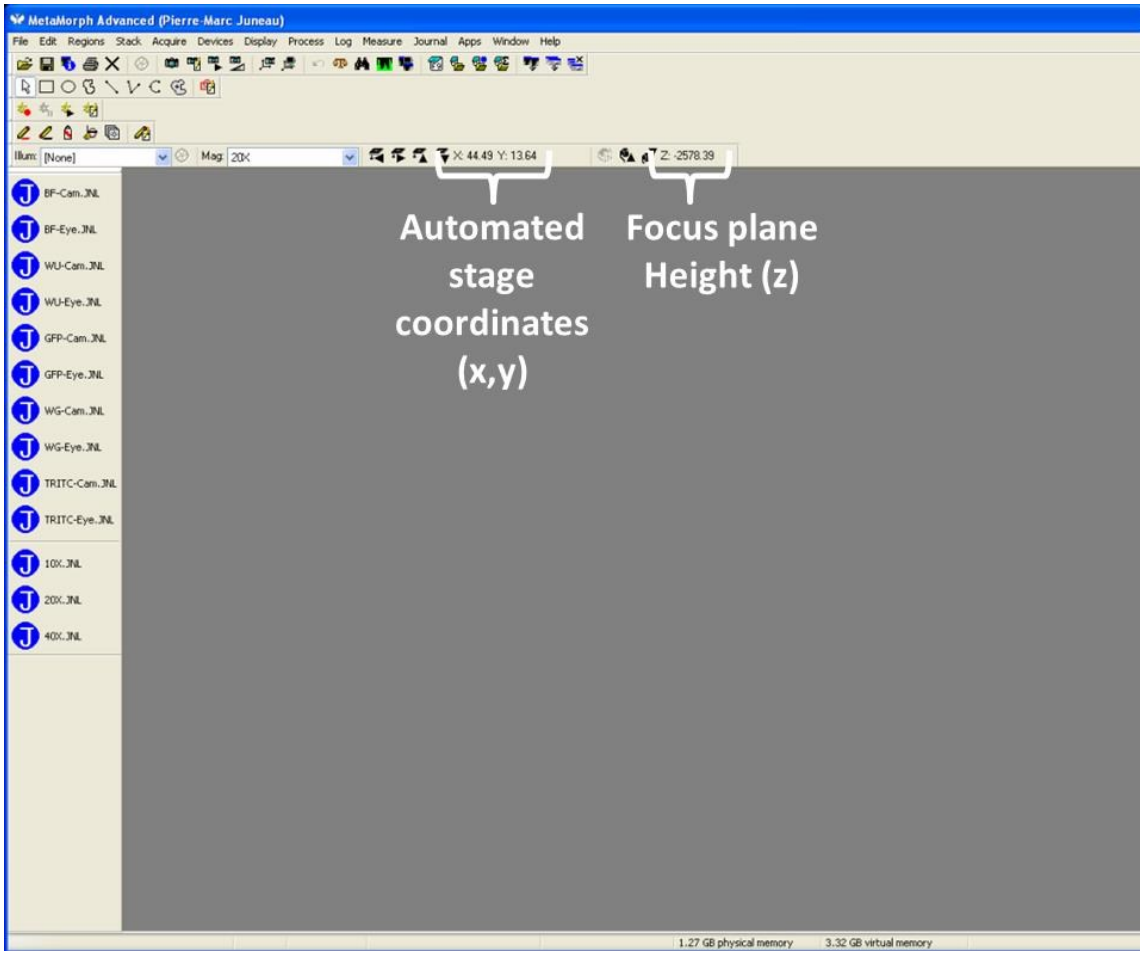


Figure E-1 : Metamorph® interface

Image acquisition is performed using the MDA (Multiple Dimension Acquisition) function of Metamorph®, which allows acquiring images at different wavelengths using several imaging set-ups (e.g. illumination) for a given list of points (x, y and z (focus)) with coordinates relative to the center of the first multiwell plate well. The MDA window can be opened in the Apps menu, as shown in Figure E-2.

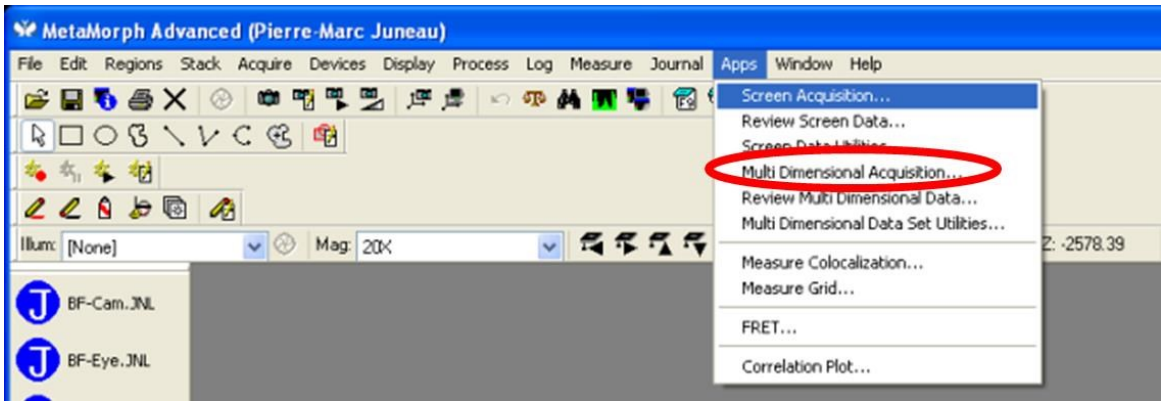


Figure E-2 : Apps menu to open MDA window

E.1 Set cell culture environment

Before bringing the cells to the microscope, ensure that the enclosure is heated (the fan and the heating element are working) and that CO_2 is flowing out of the gas controller at an appropriate concentration (the controller is ON).

E.2 Set the zero position (x=0, y=0) on the automated stage

All imaging is performed by setting the zero position of the automated stage (x=0, y=0) at the center of the first well (top left well of multiwell plate). To adjust the zero, it is possible to use en empty multiwell plate with a cross (made with a pen) marking the center of the first well. Once this empty multiwell plate is properly placed on the automated stage, use the automated stage controller joystick while looking at the image through the binoculars to reach the center of the first well (marked by a cross). Once the center of the well is reached, press the ZERO button on the automated stage controller to set the zero position.

E.3 Generate lists of points (if required)

Metamorph® (using the MDA function) will acquire images at positions that are relative to the zero position (in mm), and this is why lists of points need to be generated. To generate lists of points for each well (.STG files), a Matlab® script can be used (see POINTS_GENERATOR_METAMORPH_v0r1.m in Section E.10). The distance between each well depends on which multiwell plate is used. Roughly, these distances are 40 mm, 26 mm, 19.4 mm and 9 mm for 6 wells, 12 wells, 24 wells and 96 wells multiwell plates. The number of rows and the number of columns also depends on the multiwell plate used. The distance between each image in each well is a function of the magnification, and was set to ensure a certain overlap between images (around 30%).

For the X direction, these distances were 1.0 mm, 0.5 mm and 0.25 mm for magnifications of 10X, 20X and 40X respectively. For the Y direction, these distances were 0.8 mm, 0.4 mm and 0.2 mm for magnifications of 10X, 20X and 40X respectively. Note that the difference between the X and Y directions can be explained by the fact that the images are rectangular, and are larger in the X direction (horizontal) than in the Y direction (vertical). The X_HEIGHT and Y_HEIGHT parameters (in the Matlab® script) determine the grid to be used for imaging each well (these should be odd numbers). For example, a 3x3 grid would be performed by setting X_HEIGHT = Y_HEIGHT = 3. Note that each point is associated to a focus of zero (z=0), since the focus for each well is set manually (see Section E.6).

E.4 Program a Metamorph acquisition journal (if required)

Go to the Journal menu, and click on Edit Journal to edit a new journal (.JNL). For the acquisition of images, a series of functions needs to be programmed for each well (*i*), as illustrated in Figure E-3 and in Table E-1:

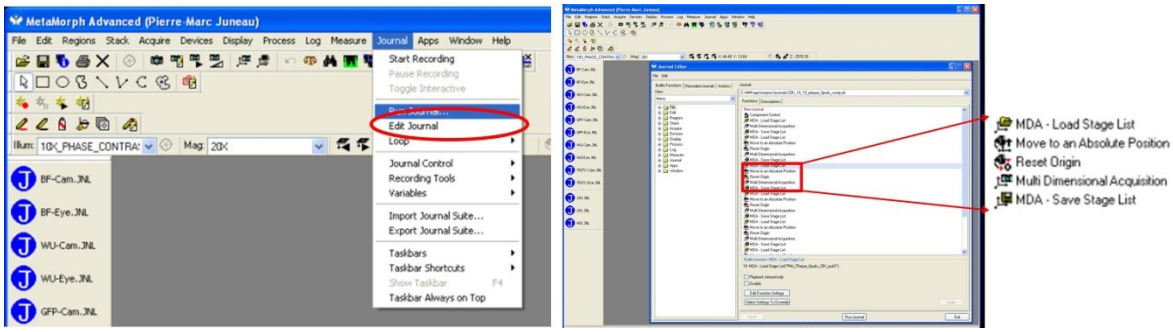


Figure E-3 : How to program a Metamorph acquisition journal

Table E-1 : Functions needed for each well *i*

Function	Description
MDA – Load Stage List	Load the list of points associated to well *i*
Move To An Absolute Position	Move the focus relative to the focus of the previous well (see Section E.6)
Reset Origin	Reset the zero focus
Multi Dimensional Acquisition	Acquire the images for each point of the list of points loaded
MDA – Save Stage List	Save the focuses achieved by the Autofocus Utility (optional)

E.5 Place the cell culture multiwell plate inside the microscope

The multiwell plate is pushed inside the secondary enclosure (see Figure 3-5); the top of the secondary enclosure is closed and is placed onto the automated stage (see Figure 3-2). The temperature probe and the CO_2 tubing are connected onto the secondary enclosure. The microscope focus and condenser height are then adjusted manually (by looking at the cells inside the first well) to ensure an optimal image quality (see manufacturer manual for condenser adjustment). Once the focus is well adjusted in the first well, run the journal RESET FOCUS to set the z axis to zero (Figure E-4). Switch manually to the side port (camera).

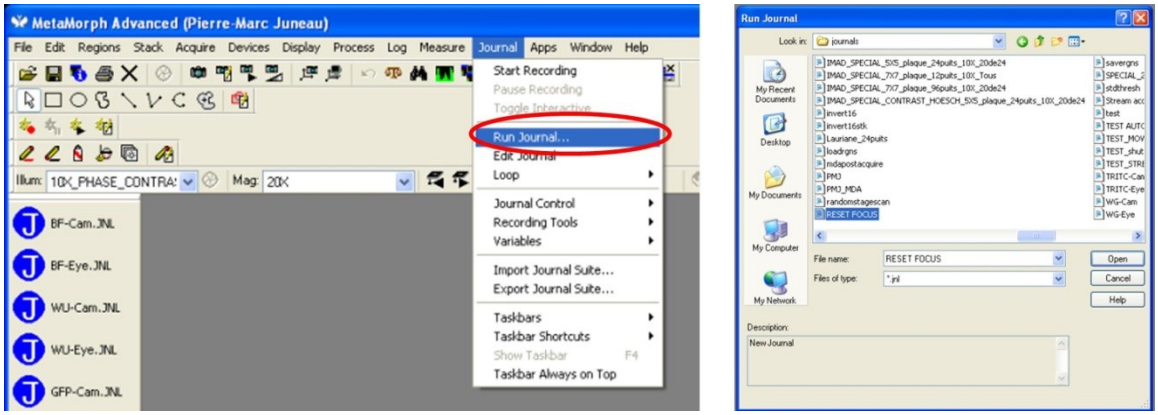


Figure E-4 : RESET FOCUS journal

E.6 Adjust manually the focus for each well

Even though the Autofocus Utility in the MDA can fine tune the focus automatically, it should be roughly adjusted manually near its optimal value for each well to avoid a focus drift. Manual adjustments are made using the Device Control window while looking at the cells using the MDA (live acquisition button, circled in red), as can be seen in Figure E-5. It is possible to go from well center to well center by entering the Stage X and Stage Y coordinates for each well, and to adjust the focus (z) until it is optimal (note the optimal value for each well). A table can be constructed in Excel®, computing the difference between the focuses of two consecutive wells (example provided in Figure E-6), and differences are entered in the *Move To An Absolute Position* function for each well (see Section E.7).

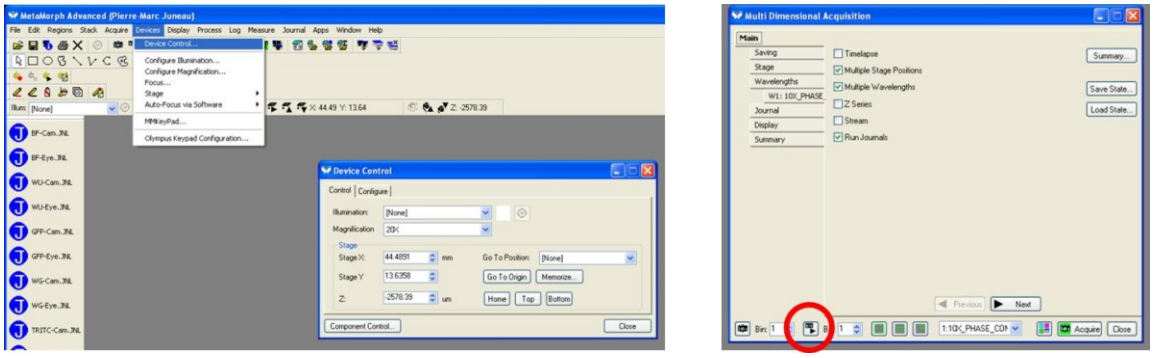
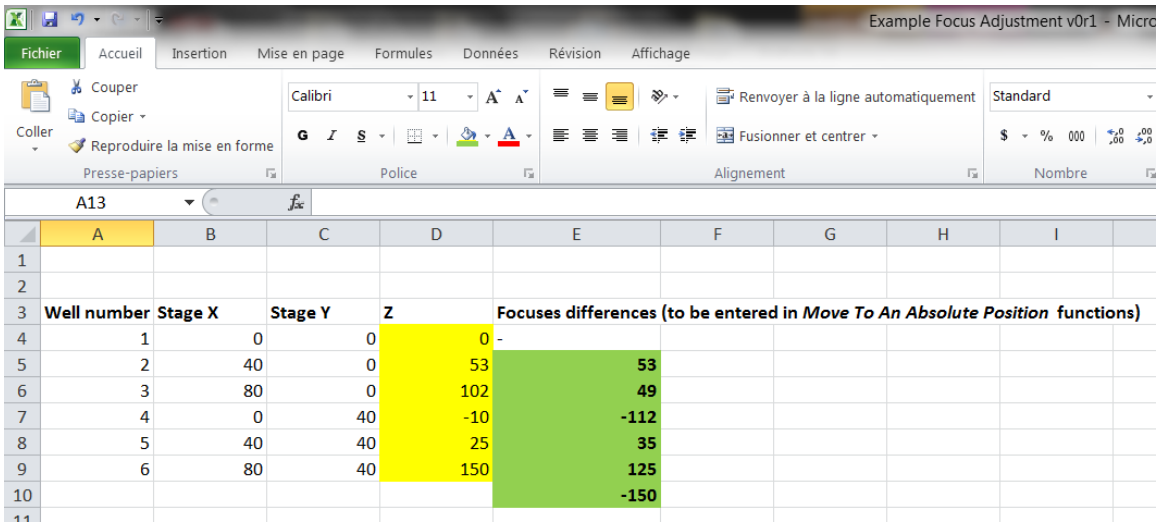


Figure E-5 : Device Control and MDA windows (live acquisition) to adjust manually the focus for each well



Well number	Stage X	Stage Y	Z	Focuses differences (to be entered in *Move To An Absolute Position* functions)
1	0	0	0	-
2	40	0	53	53
3	80	0	102	49
4	0	40	-10	-112
5	40	40	25	35
6	80	40	150	125
				-150

Figure E-6 : Example of focuses differences computed between adjacent wells for a 6 wells multiwell plate

E.7 Enter the proper values in the "*Move To An Absolute Position*" functions

Starting from the second well, the *Move To An Absolute Position* functions (see Figure E-7) are set to the values of focuses differences obtained in Section E.6.

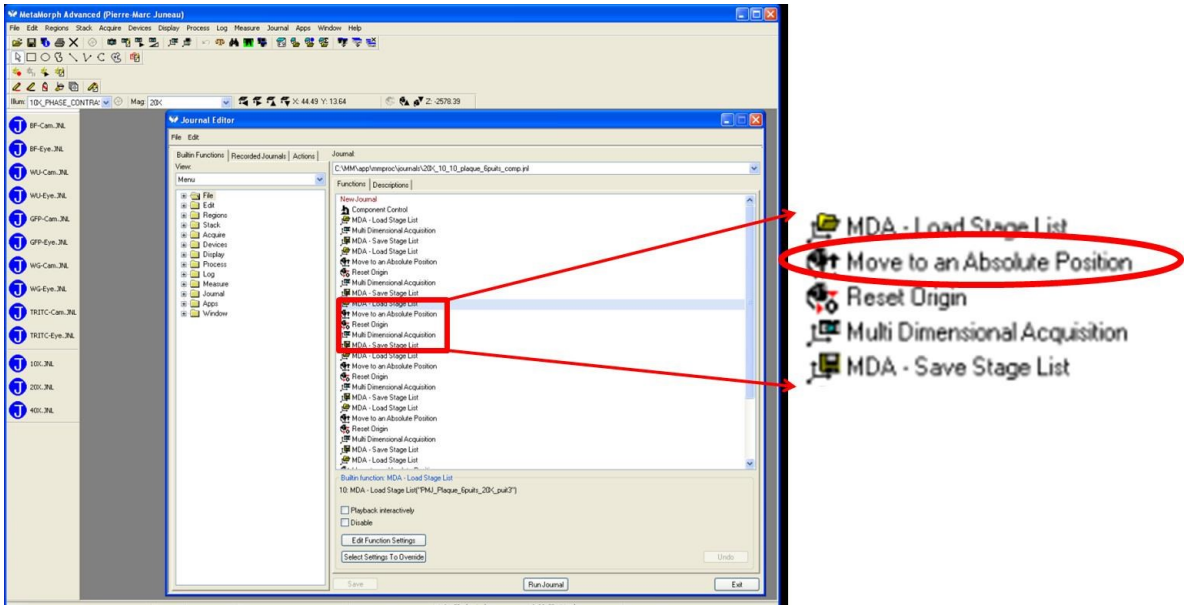


Figure E-7 : The *Move to an Absolute Position* value (starting from well #2) needs to be modified to incorporate the focuses differences obtained in Section E.6.

E.8 Set all the settings in the MDA window

Before starting the acquisition, different options of the MDA window (Figure E-8) need to be set:

- ***Saving***: A meaningful base name should be used (e.g. cell phenotype, passage, etc.). End the name with "_1": this number will be incremented for each well and each acquisition. Choose also the folder in which the images will be saved (usually an external hard disk).
- ***Wavelengths***: For each position in the points list, it is possible to capture several wavelengths (e.g. phase contrast + epifluorescence with different filter cubes). Also, the Autofocus Utility settings can be adjusted in this window. For this PhD project the Autofocus Utility was ran for each image acquired in phase contrast microscopy to optimize focus adjustment (for textural analysis).

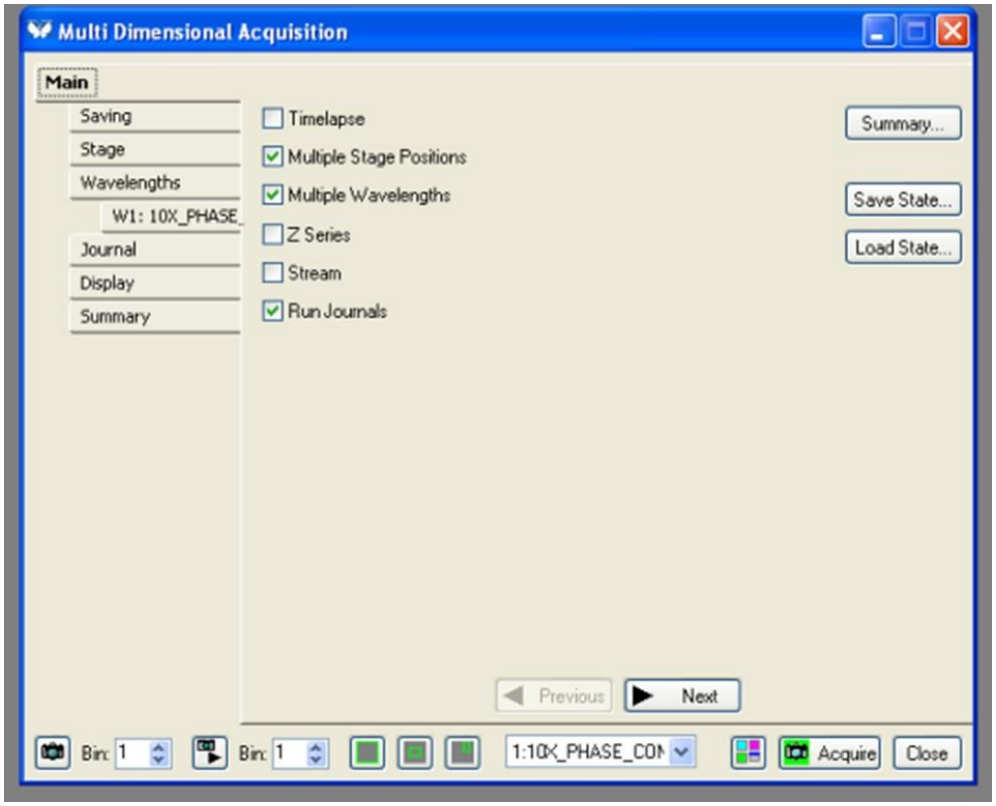


Figure E-8 : MDA window with different options/settings

E.9 Launch the image acquisition

To perform live cell imaging for several hours or days, it is possible to use a loop in a journal (MDA journal) to run the image acquisition journal at fixed intervals. For example (Figure E-9), the *PMJ_MDA* journal is set to run every three hours (image all points in all the wells) for 300 times. However, it is preferable to stop and restart the acquisition once a day because of memory limitations (Metamorph® could crash). Image acquisition can be started by running the MDA journal. An example of an image typical name structure is provided in Figure E-10.

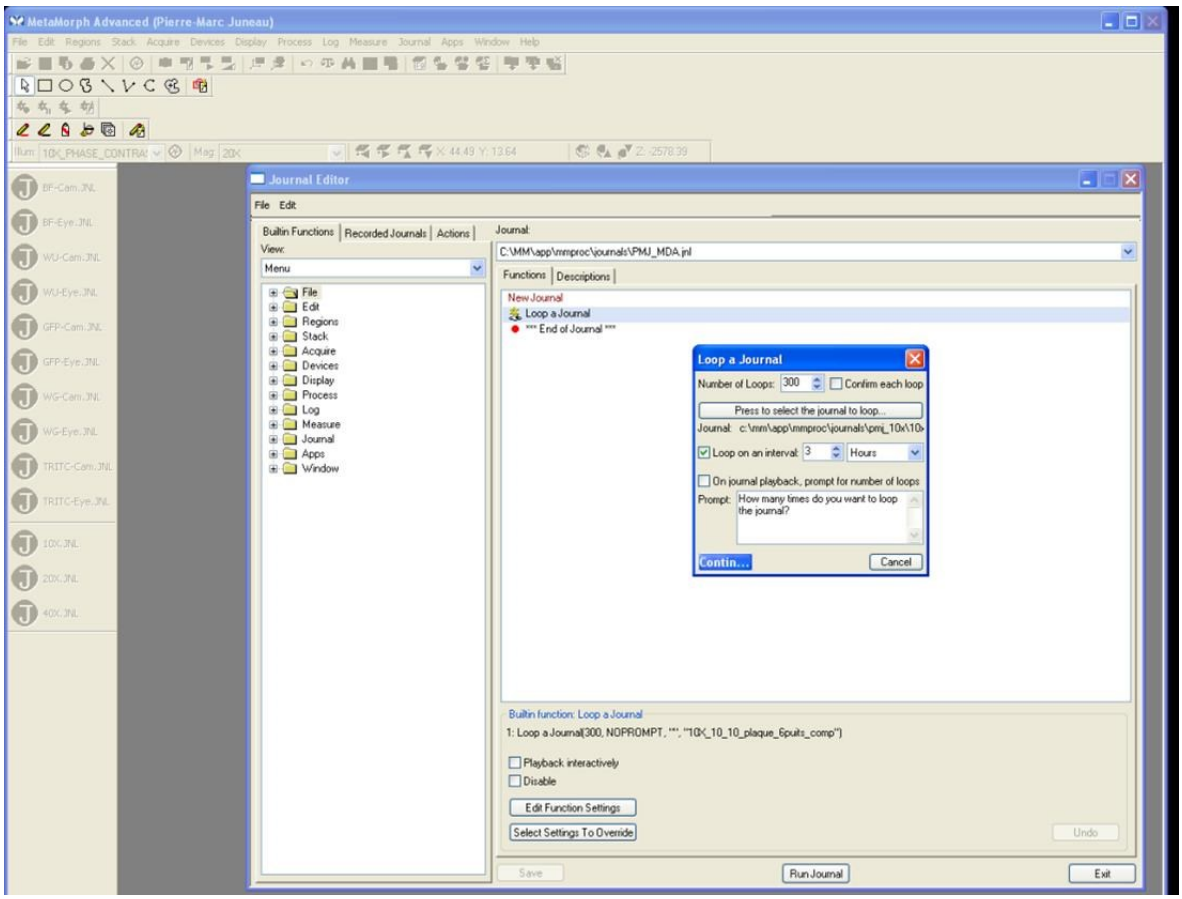


Figure E-9 : MDA journal for image acquisition at fixed intervals

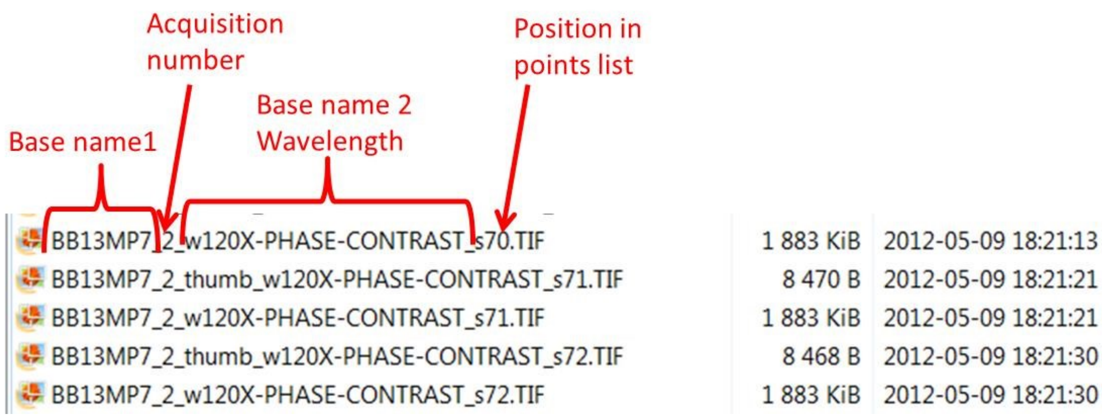


Figure E-10 : Examples of images acquired (typical name structure)

E.10 Matlab® script to generate lists of points for MDA (Metamorph®)

POINTS_GENERATOR_METAMORPH_v0r1.m

```
%This m-file creates a text file to load points in Metamorph. The points
%form a grid. There is one grid (one text file) per well.

%The .STG files must be put in C:\MM\app\mmproc\DATA

%Those files can be used with the MDA application in Metamorph (for
%scanning multiwell plates).

%IMPORTANT NOTE: The position (0,0) of the stage must be set at the center
%of the first well (top left corner).


%%Author: Pierre-Marc Juneau
%%Date: August 2010

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%INPUTS%%

%Basename for the .STG files

BASENAME='JPL_40X_LIST_Plaque_24puits_puit_';

%Intervals (in mm) between images (depending on focus):
%10X: Delta_X=1.0, Delta_Y=0.8
```

```
%20X: Delta_X=0.5, Delta_Y=0.4
%40X: Delta_X=0.25, Delta_Y=0.2

Delta_X=0.25;
Delta_Y=0.2;
Corners=-2; %The Corners option could be used to "chop" some unwanted points at the
corners

%Distances between wells (depending on type of multiwell plate, in mm):
%6 well plate: 40
%12 well plate: 26
%24 well plate: 19.4
%96 well plate: 9.0

Bias_X=19.4;
Bias_Y=Bias_X;

%Number of rows and columns in the multiwell plate:

n_Rows=4;
n_Colums=6;

%Grid dimensions (depending on acquisition time wanted):
%For example, for an 11x11 grid, we have X_HEIGHT=11 and Y_HEIGHT=11
%Note that those must be odd numbers!

X_HEIGHT=3;
Y_HEIGHT=3;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initialisation of the focus matrix
Number_points=X_HEIGHT*Y_HEIGHT;
z=zeros(Number_points,1);

nX=floor(X_HEIGHT/2);
nY=floor(Y_HEIGHT/2);

HEIGHT_X=nX*Delta_X;
HEIGHT_Y=nY*Delta_Y;

Well_number=1;

%For loop for printing the files

for ii=1:n_Rows
    for jj=1:n_Colums
```

```
BiasX=(jj-1)*Bias_X;
BiasY=(ii-1)*Bias_Y;

pos_y_high=HEIGHT_Y+BiasY;
pos_y_low=-HEIGHT_Y+BiasY;
pos_x_high=HEIGHT_X+BiasX;
pos_x_low=-HEIGHT_X+BiasX;


filename=[BASENAME num2str(Well_number) '.STG']

fid=fopen(filename,'wt');

fprintf(fid,'"Stage Memory List", Version 5.0\n');
fprintf(fid,'0, 0, 0, 0, 0, 0, 0, "mm", "mm"\n');
fprintf(fid,'0\n');
fprintf(fid,'%0.0f\n',Number_points);

a=1;

for i=pos_y_high:-Delta_Y:pos_y_low
    for j=pos_x_low:Delta_X:pos_x_high

        a
        perm1=0;
        perm2=0;

        if i<(pos_y_high-Corners) && i>(pos_y_low+Corners)
            perm1=1;
        end
        if j<(pos_x_high-Corners) && j>(pos_x_low+Corners)
            perm2=1;
        end
        if perm1==1 || perm2==1;
        fprintf(fid,'"%0.0f", %0.2f, %0.2f, %0.2f, 0, 0, FALSE, -9999, TRUE, TRUE,
0\n',a,j,i,z(a));
         a=a+1;
        end
    end
end
fclose(fid);
Well_number=Well_number+1;
    end
end
```

Appendix F. Matlab® scripts used for image and data processing

Metamorph_IMAGE_CALIBRATION_v0r2.m

```
%This m-file is for having a calibration image for segmentation (to generate a
calibration segmentation map)

%Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%


I_M=imread('21.tif');

%I_M=imread('LOBSFM_Basse_densité2.tif');
%I_M=imread('LOBSFM_Medium_densité.tif');
%I_M=imread('LOBSFM_Haute_Densité.tif');

%I_M=imread('Serum_Basse_densité.tif');
%I_M=imread('Serum_Basse_densité2.tif');
%I_M=imread('Serum_Haute_densité.tif');


criteria=1;

[a,b] = size(I_M);

Mat=zeros(a,b);


i=1;
%Mat=cell2mat(struct2cell(load('21')));
%i=max(max(Mat))+1;

while criteria==1

    MAT_IMAGE=label2rgb(Mat,'jet',[.5 .5 .5]);
    figure(1),
    imshow(MAT_IMAGE)
    hold on
    h = imshow(I_M);
    set(h, 'AlphaData', 0.9);

    B = roipoly;



    INPUT=input('Do you want to continue? 1-yes 0-no 2-Not satisfied: want to go back');

    if INPUT==0
        criteria=0;
        Mat(B==1)=i;
        i=i+1;
    end

    if INPUT==1
        Mat(B==1)=i;
        i=i+1;
    end
    Savefile_name=['21'];
save(Savefile_name,'Mat');
end
```

Metamorph_TEST_CELL_CLUSTERS_AUTOMATED_COMPUTATION_TIME_ALL_v2.m

```
%This function is used to calculate computing times required for each
%method

%Pierre-Marc Juneau, November 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
clear
clc
%Choose method
RANGE_FILTER=1;
STANDARD_DEVIATION_FILTER=0;
ENTROPY_FILTER=0;
Topman=0;
Dewan=0;
Huth=0;
Bradhurst=0;
Debeir=0;

FIGURES_OUTPUT=1;


%%
%Get the directory where the images are
directory_name_image = uigetdir('','Choose the directory where the images are');
listing_image = dir([directory_name_image,'\','*.tif']);
[Number_of_images,b]=size(listing_image);

%Get the directory where the calibration matrix are
directory_name_cal  =  uigetdir('','Choose  the  directory  where  the  calibration  matrices
are');
listing_cal = dir(directory_name_cal);
[Number_of_cal,b]=size(listing_cal);



IMAGES_NAME_LIST={listing_image.name}';

DATA_GLOBAL=[];

DATA_FIXED_PARAMETERS=[];


alpha=0.5;


Recall_AVERAGE=0;
Precision_AVERAGE=0;
Berkeley_AVERAGE=0;
ERROR_SURFACE_AVERAGE=0;

for m=1:Number_of_images
    m
    IMAGE_NAME=listing_image(m).name;
    MATRIX_NAME=strrep(IMAGE_NAME,'.TIF', '');

    I_M=imread([directory_name_image,'\',IMAGE_NAME]);

    Calibration=cell2mat(struct2cell(load([directory_name_cal,'\',MATRIX_NAME])));

    Calibration(Calibration>0)=1;

    [a,b] = size(I_M);


    if RANGE_FILTER==1
        cut_cells_regions=3000;
        filter_size=3;
        Thresh=10;
```

```
        tic;

[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_RANGE_FILTER_v0r1(I_M,filter_size,Th
resh,cut_cells_regions);
        TIME_COMPUTATION=toc;
    end

    if STANDARD_DEVIATION_FILTER==1
        cut_cells_regions=3000;
        filter_size=7;
        Thresh=6;

        tic;

[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_STDFILTER_v0r1(I_M,filter_size,Thres
h,cut_cells_regions);
        TIME_COMPUTATION=toc;

    end

    if ENTROPY_FILTER==1
        cut_cells_regions=4000;
        filter_size=7;
        Thresh=4;

        tic;

[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_ENTROPYFILTER_v0r1(I_M,filter_size,T
hresh,cut_cells_regions);
        TIME_COMPUTATION=toc;


    end


    if Topman==1
        bigWindow=29;
        smallWindow=1;
        Thresh=0.05;
        tic;
        [confluency,outImage]=Topman_2011(I_M,bigWindow,smallWindow,Thresh);
        I_bin_regions=~outImage;
        TIME_COMPUTATION=toc;

    end

    if Dewan==1
        r=67;
        h=53;
        tic;
        I_bin_regions=Dewan_2011_v0r1(I_M,r,h);
        TIME_COMPUTATION=toc;

    end

    if Huth==1
        FS_entropy=25;
        FS_median=53;
        Thresh_particles=8000;
        tic;
        I_bin_regions=Huth_method_v0r1(I_M,FS_entropy,FS_median,Thresh_particles);
        TIME_COMPUTATION=toc;

    end


    if Bradhurst==1
        filter_size=7;
        Thresh_particles=3500;
        Thresh_holes=10000;
```

```
        tic;

I_bin_regions=Bradhurst_modified_method_v0r1(I_M,filter_size,Thresh_particles,Thresh_hole
s);
        TIME_COMPUTATION=toc;

    end

     if Debeir==1
        filter_size=27;
        Thresh=19;
        tic;
        [I_bin_regions]=Debeir_2008_v0r1(I_M,filter_size,Thresh);
        TIME_COMPUTATION=toc;

     end

     if FIGURES_OUTPUT==1

         IMAGE_TO_SAVE=label2rgb(I_M,'Gray');

         [r,s]=size(I_M);

         for f=1:r
             for g=1:s
                 if I_bin_regions(f,g)>0 && Calibration(f,g)>0
                     IMAGE_TO_SAVE(f,g,2)=255;
                 end
                 if I_bin_regions(f,g)>0 && Calibration(f,g)==0
                     IMAGE_TO_SAVE(f,g,3)=255;
                 end
                 if I_bin_regions(f,g)==0 && Calibration(f,g)>0
                     IMAGE_TO_SAVE(f,g,1)=255;
                 end
                 if I_bin_regions(f,g)==0 && Calibration(f,g)==0
                 end
             end
         end

         save_name=[MATRIX_NAME,'.jpg'];
         imwrite(IMAGE_TO_SAVE,save_name,'jpg');

     end


    %Recall and precision according to He 2007

    A_U_M=sum(sum((double(I_bin_regions)).*(Calibration)));
    M=sum(sum(Calibration));
    A=sum(sum((double(I_bin_regions))));
    Recall=A_U_M/M;
    Precision=A_U_M/A;


    %F-Berkeley according to Al-Muhairi 2011

    F_BERKELEY=(Recall*Precision)/(alpha*Recall+(1-alpha)*Precision);

    %Error relative to surface recovered

    ALGORITHM_SURFACE=sum(sum(double(I_bin_regions)));
    CALIBRATION_SURFACE=sum(sum(double(Calibration)));

    ERROR_SURFACE=abs(CALIBRATION_SURFACE-ALGORITHM_SURFACE)/CALIBRATION_SURFACE*100;

    %%%%%%

    DATA_INTERN=[Recall,Precision,F_BERKELEY,ERROR_SURFACE,TIME_COMPUTATION];

    DATA_FIXED_PARAMETERS=[DATA_FIXED_PARAMETERS;DATA_INTERN];
```

```
    Recall_AVERAGE=Recall_AVERAGE+Recall;
    Precision_AVERAGE= Precision_AVERAGE+Precision;
    Berkeley_AVERAGE=Berkeley_AVERAGE+F_BERKELEY;
    ERROR_SURFACE_AVERAGE=ERROR_SURFACE_AVERAGE+ ERROR_SURFACE;


end


HEADLINE={'Image  Number'  'Recall'  'Precision'  'F-Berkeley'  'Average  Surface  Error'
'Computation time (s)'};

xlswrite('Computation_Performance_',HEADLINE,1,'A1');

xlswrite('Computation_Performance_',IMAGES_NAME_LIST,1,'A2');

xlswrite('Computation_Performance_',DATA_FIXED_PARAMETERS,1,'B2');
```

Bradhurst_modified_method_v0r1.m

```
%Method adapted from...

%Christopher J. Bradhurst, Wageeh Boles, Yin Xiao, Segmentation of Bone Marrow Stromal
Cells in Phase
%Contrast Microscopy Images, IEEE 2008

%Programmed by Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%INPUTS:
%'I_M' is the raw image

%'display' in a local input for displaying graphics

%OUTPUTS:
%DATA_OUTPUT contains the background and cell clusters statistics

%VARIABLES_NAMES contains the name of the variables

%%
function
I_bin_out=Bradhurst_modified_method_v0r1(I_M,filter_size,Thresh_particles,Thresh_holes)

%%
%Segmenting cellular regions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1-Filtering
I_filtered = stdfilt(I_M,ones(filter_size,filter_size));

%2-Thresholding
[imagBW,th]=kittlerMet((I_filtered));


%3-Particle filters
REGIONS_LABELED_particles=bwlabeln(imagBW,4);
REGIONS_PROPS_particles=regionprops(REGIONS_LABELED_particles,'Area');
I_bin_regions_particles=(ismember(REGIONS_LABELED_particles,find([REGIONS_PROPS_particles
.Area]>=Thresh_particles)));

INVERTED_BW=~imagBW;

REGIONS_LABELED_blackholes=bwlabeln(INVERTED_BW,4);
REGIONS_PROPS_blackholes=regionprops(REGIONS_LABELED_blackholes,'Area');
I_bin_regions_blackholes=(ismember(REGIONS_LABELED_blackholes,find([REGIONS_PROPS_blackho
les.Area] < Thresh_holes)));

I_bin_out=I_bin_regions_particles;

I_bin_out(I_bin_regions_blackholes==1)=1;
```

Debeir_2008_v0r1.m

```
%Programmed by Pierre-Marc Juneau, November 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
function [I_bin_regions]=Debeir_2008_v0r1(I_M,filter_size,Thresh)

%Filter for 20% percentile

order_20=floor(filter_size^2*0.2);

I_20=ordfilt2(I_M,order_20,ones(filter_size,filter_size),'symmetric');


%Filter for 80% percentile

order_80=floor(filter_size^2*0.8);

I_80=ordfilt2(I_M,order_80,ones(filter_size,filter_size),'symmetric');


%Morphological Gradient

I_bin_regions=I_80-I_20;

%Thresholding the morphological gradient

I_bin_regions(Thresh>I_bin_regions)=0;
I_bin_regions(I_bin_regions>0)=1;
```

Dewan_2011_v0r1.m

```
%Programmed by Pierre-Marc Juneau, November 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
function [I_bin_regions]=Dewan_2011_v0r1(I_M,r,h)


se = strel('ball',r,h);

I_inverted=double(imcomplement(I_M));

I_filtered=imtophat(I_inverted,se);


[I_bin_regions,th]=otsu((I_filtered));
```

Huth_method_v0r1.m

```
%Method adapted from...

%Huth 2010

%Programmed by Pierre-Marc Juneau, November 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%INPUTS:
%'I_M' is the raw image

%'display' in a local input for displaying graphics

%OUTPUTS:
%DATA_OUTPUT contains the background and cell clusters statistics

%VARIABLES_NAMES contains the name of the variables

%%
function I_bin_out=Huth_method_v0r1(I_M,FS_entropy,FS_median,Thresh_particles)

%%
%Segmenting cellular regions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1-Filtering
I_filtered_entropy = entropyfilt(I_M,ones(FS_entropy,FS_entropy));

I_filtered_median = medfilt2(I_M,[FS_median FS_median]);

%2-Multiplication and thresholding

I_multiplied=double(I_filtered_entropy).*double(I_filtered_median);

[imagBW,th]=otsu((I_multiplied));


%3-Particle filters
REGIONS_LABELED_particles=bwlabeln(imagBW,4);
REGIONS_PROPS_particles=regionprops(REGIONS_LABELED_particles,'Area');
I_bin_regions_particles=(ismember(REGIONS_LABELED_particles,find([REGIONS_PROPS_particles
.Area] >= Thresh_particles)));

I_bin_out=I_bin_regions_particles;
```

Metamorph_CELL_CLUSTERS_SEGMENTATION_ENTROPYFILTER_v0r1.m

```
%This function aims at roughly segmenting the cellular regions (cell
%clusters) from the background. This method is different but similar to the method of:

%Christopher J. Bradhurst, Wageeh Boles, Yin Xiao, Segmentation of Bone Marrow Stromal
Cells in Phase
%Contrast Microscopy Images, IEEE 2008

%Programmed by Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%INPUTS:
%'I_M' is the raw image

%'display' in a local input for displaying graphics

%OUTPUTS:
%DATA_OUTPUT contains the background and cell clusters statistics

%VARIABLES_NAMES contains the name of the variables

%%
%function
[I_out1,I_bin_regions,AVERAGE_BACKGROUND,STD_BACKGROUND,DATA_OUTPUT,VARIABLES_NAMES]=Meta
morph_CELL_CLUSTERS_SEGMENTATION_v0r1(I_M)
function
[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_ENTROPYFILTER_v0r1(I_M,filter_size,T
hresh,cut_cells_regions)

%%
%Segmenting cellular regions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I_bin_regions = entropyfilt(I_M,ones(filter_size,filter_size));
RANGE_FILT_HIST=I_bin_regions;

I_bin_regions(Thresh>I_bin_regions)=0;
I_bin_regions(I_bin_regions>0)=1;

I_bin_regions=imfill(I_bin_regions);

%%
%Filter pixel groups too small
REGIONS_LABELED1=bwlabeln(I_bin_regions,4);
REGIONS_PROPS1=regionprops(REGIONS_LABELED1,'Area');
I_bin_regions=uint8(ismember(REGIONS_LABELED1,find([REGIONS_PROPS1.Area]                >=
cut_cells_regions)));
```

Metamorph_CELL_CLUSTERS_SEGMENTATION_RANGE_FILTER_v0r1.m

```
%This function aims at roughly segmenting the cellular regions (cell
%clusters) from the background. This method is different but similar to the method of:

%Christopher J. Bradhurst, Wageeh Boles, Yin Xiao, Segmentation of Bone Marrow Stromal
Cells in Phase
%Contrast Microscopy Images, IEEE 2008

%Programmed by Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%
function
[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_RANGE_FILTER_v0r1(I_M,filter_size,Th
resh,cut_cells_regions)



%%
%Segmenting cellular regions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I_bin_regions = rangefilt(I_M,ones(filter_size,filter_size));

%Thresholding

I_bin_regions(Thresh>I_bin_regions)=0;
I_bin_regions(I_bin_regions>0)=1;

I_bin_regions=imfill(I_bin_regions);

%%
%Filter pixel groups too small
REGIONS_LABELED1=bwlabeln(I_bin_regions,4);
REGIONS_PROPS1=regionprops(REGIONS_LABELED1,'Area');
I_bin_regions=uint8(ismember(REGIONS_LABELED1,find([REGIONS_PROPS1.Area]                >=
cut_cells_regions)));
```

Metamorph_CELL_CLUSTERS_SEGMENTATION_STDFILTER_v0r1.m

```
%This function aims at roughly segmenting the cellular regions (cell
%clusters) from the background. This method is different but similar to the method of:

%Christopher J. Bradhurst, Wageeh Boles, Yin Xiao, Segmentation of Bone Marrow Stromal
Cells in Phase
%Contrast Microscopy Images, IEEE 2008

%Programmed by Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%INPUTS:
%'I_M' is the raw image

%'display' in a local input for displaying graphics

%OUTPUTS:
%DATA_OUTPUT contains the background and cell clusters statistics

%VARIABLES_NAMES contains the name of the variables

%%
function
[I_bin_regions]=Metamorph_CELL_CLUSTERS_SEGMENTATION_STDFILTER_v0r1(I_M,filter_size,Thres
h,cut_cells_regions)


%%
%Segmenting cellular regions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I_bin_regions = stdfilt(I_M,ones(filter_size,filter_size));

I_bin_regions(Thresh>I_bin_regions)=0;
I_bin_regions(I_bin_regions>0)=1;


I_bin_regions=imfill(I_bin_regions);

%%
%Filter pixel groups too small
REGIONS_LABELED1=bwlabeln(I_bin_regions,4);
REGIONS_PROPS1=regionprops(REGIONS_LABELED1,'Area');
I_bin_regions=uint8(ismember(REGIONS_LABELED1,find([REGIONS_PROPS1.Area]                >=
cut_cells_regions)));
```

Topman_2011.m

```
%Cell-covered surface for Topman's algorithm

function [confluency,outImage]=Topman_2011(inImage,bigWindow,smallWindow,threshold)

%Convert image to grayscale and
% adjust the histogram
if (size(inImage, 3) == 3)
inImage = rgb2gray(inImage);
end

inImage = mat2gray(inImage);
% Small window flow
smallStd = stdfilt(inImage, ...
ones([smallWindow smallWindow]));
smallThresh = ~im2bw(smallStd, threshold);

% Big window flow
bigStd = stdfilt(inImage, ...
ones([bigWindow bigWindow]));
bigThresh = ~im2bw(bigStd, threshold);
bigDilation = imdilate(bigThresh,ones([floor(bigWindow/2) floor(bigWindow/2)]));

% Final flow
intersectImage = bigDilation .* smallThresh;
closeImage = imclose(intersectImage, ...
ones([smallWindow smallWindow]));
outImage = imopen(closeImage, ...
ones([smallWindow smallWindow]));

% Calculate confluency
confluency = 100*(1-mean(outImage(:)));
end
```

otsu.m

```
function [imagBW,th] = otsu(imag)
% OTSU binarizes a gray scale image 'imag' into a binary image, with the
% noises removed.
% Input:
%   imag: the gray scale image, with black foreground(0), and white
%   background(255).
% Output:
%   imagBW: the binary image of the gray scale image 'imag', with Otsu
%   algorithm.

% Reference:
%   Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms.
%   IEEE Transactions on Systems, Man, and Cybernetics. 1979.SMC-9(1):62-66

resolution=256;
%%



imag = imag(:, :, 1);

%scaling of image
imag=uint8((imag-min(min(imag)))./(max(max(imag))-min(min(imag))).*255);


%[a,b]=size(imag);

%imag2=reshape(imag,a*b,1);

%[counts, x] = hist(imag2,resolution); % counts are the histogram. x is the intensity
level.
%x=x';
%counts=counts';
[counts, x] = imhist(imag);
GradeI = length(x);   % the resolusion of the intensity. i.e. 256 for uint8.
varB = zeros(GradeI, 1);  % Between-class Variance of binarized image.

prob = counts ./ sum(counts);  % Probability distribution
meanT = 0;  % Total mean level of the picture
for i = 0 : (GradeI-1)
    meanT = meanT + i * prob(i+1);
end
varT = ((x-meanT).^2)' * prob;
% Initialization
w0 = prob(1);   % Probability of the first class
miuK = 0;   % First-order cumulative moments of the histogram up to the kth level.
varB(1) = 0;
% Between-class variance calculation
for i = 1 : (GradeI-1)
    w0 = w0 + prob(i+1);
    miuK = miuK + i * prob(i+1);
    if (w0 == 0) || (w0 == 1)
        varB(i+1) = 0;
    else
        varB(i+1) = (meanT * w0 - miuK) .^ 2 / (w0 * (1-w0));
    end
end

maxvar = max(varB);
em = maxvar / varT;  % Effective measure
index = find(varB == maxvar);
index = mean(index);
th = (index-1)/(GradeI-1);
imagBW = im2bw(imag, th);

% thOTSU = graythresh(imag)
% imagBWO = im2bw(imag, thOTSU);
```

kittlerMet.m

```
function [imagBW,th] = kittlerMet(imag)
% KITTLERMET binarizes a gray scale image 'imag' into a binary image
% Input:
%   imag: the gray scale image, with black foreground(0), and white
%   background(255).
% Output:
%   imagBW: the binary image of the gray scale image 'imag', with kittler's
%   minimum error thresholding algorithm.

% Reference:
%   J. Kittler and J. Illingworth. Minimum Error Thresholding. Pattern
%   Recognition. 1986. 19(1):41-47

MAXD = 100000;
imag = imag(:,:,1);


%scaling of image
imag=uint8((imag-min(min(imag)))./(max(max(imag))-min(min(imag))).*255);




[counts, x] = imhist(imag);  % counts are the histogram. x is the intensity level.
GradeI = length(x);   % the resolusion of the intensity. i.e. 256 for uint8.
J_t = zeros(GradeI, 1);  % criterion function
prob = counts ./ sum(counts);  % Probability distribution
meanT = x' * prob;  % Total mean level of the picture
% Initialization
w0 = prob(1);   % Probability of the first class
miuK = 0;   % First-order cumulative moments of the histogram up to the kth level.
J_t(1) = MAXD;
n = GradeI-1;
for i = 1 : n
    w0 = w0 + prob(i+1);
    miuK = miuK + i * prob(i+1);  % first-order cumulative moment
    if (w0 == 0) || (w0 == 1)
        J_t(i+1) = MAXD;    % T = i
    else
        miu1 = miuK / w0;
        miu2 = (meanT-miuK) / (1-w0);
        var1 = (((0 : i)'-miu1).^2)' * prob(1 : i+1);
        var1 = var1 / w0;  % variance
        var2 = (((i+1 : n)'-miu2).^2)' * prob(i+2 : n+1);
        var2 = var2 / (1-w0);
        if var1 > 0 && var2 > 0   % in case of var1=0 or var2 =0
            J_t(i+1)  =  1+w0  *  log(var1)+(1-w0)  *  log(var2)-2*w0*log(w0)-2*(1-w0)*log(1-
w0);
        else
            J_t(i+1) = MAXD;
        end
    end
end
minJ = min(J_t);
index = find(J_t == minJ);
th = mean(index);
th = (th-1)/n;
imagBW = im2bw(imag, th);

% figure, imshow(imagBW), title('kittler binary');
```

Metamorph_IMAGE_NAME_STRUCTURE_v0r5.m

```
%This m-file aims at getting the names of all the images in an experiment
%using the MDA application from Metamorph

%Pierre-Marc Juneau, August 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
clear
MAX_ACQUISITION=1000;
%Number_of_image_per_acquisition=121;

%

Experience_name=input('What is the experience name?:');
Number_of_image_per_acquisition=input('How many acquisition per well ?:');
Number_of_wells=input('How many wells in multiwell plate?:');

Passage=[];
Medium=[];

for jj=1:Number_of_wells

    INTER_REPONSE=input(['What passage for well #' num2str(jj) '?:']);
    INTER_REPONSE2=input(['What medium for well #' num2str(jj) '?:']);

    Passage=[Passage;INTER_REPONSE];
    %Medium{1,jj}=INTER_REPONSE2;
    Medium=[Medium cellstr(INTER_REPONSE2)];


end

MEDIUM_OUT{1,1}=Medium;


%Getting the folder where the pictures are
directory_name = uigetdir('','Choose the directory where the image folders are');

listing = struct2cell(dir(directory_name))';

[Number_of_folders,b]=size(listing);


 % 1=sort by name, 2=sort by date/time
 listing = sortrows(listing, 1)';


%%
%For the images in each folder

Folder_TOTAL=[];
BASE_NAME1_TOTAL=[];
BASE_NAME2_TOTAL=[];
MIN_TOTAL=[];
MAX_TOTAL=[];
TIME_TOTAL=[];
DATA=[];

for i=1:Number_of_folders
    display(listing(1,i));
    INPUT=input('Do you want to include this folder?:  1-Yes 0-No ');

    if INPUT==1

        FULL_FOLDER_NAME=[directory_name,'\',char(listing(1,i))];

        %Finding the minimum value

        NAMES=[FULL_FOLDER_NAME,'\','*.nd'];
```

```
listing2=dir(NAMES);
[g,h]=size(listing2);

ACQUISITION_NUMBER=[];
for k=1:g
    FILE_NAME=strrep(listing2(k).name, '.nd', '');
    ID_VECTOR=isstrprop(FILE_NAME, 'digit');
    [u,v]=size(ID_VECTOR);

    if ID_VECTOR(v)==0
        MIN_LOCAL=0;
    else

    criteria=1;
    number=v;
    while criteria==1
        if ID_VECTOR(number)==1
            number=number-1;
        else
            criteria=0;
            number=number+1;
        end
    end

    MIN_LOCAL=str2num(FILE_NAME(number:v));
    end

    ACQUISITION_NUMBER=[ACQUISITION_NUMBER MIN_LOCAL];

end

MIN=min(ACQUISITION_NUMBER);

%Getting BASE_NAME_1

if MIN==0
    TEMP=[FULL_FOLDER_NAME,'\','*','.nd'];
    TEMP2=dir(TEMP);
    BASE_NAME1=TEMP2.name;
    string_to_remove=['.nd'];
else
    TEMP=[FULL_FOLDER_NAME,'\','*',num2str(MIN),'.nd'];
    TEMP2=dir(TEMP);
    BASE_NAME1=TEMP2.name;
    string_to_remove=[num2str(MIN),'.nd'];
end
BASE_NAME1=strrep(BASE_NAME1,string_to_remove, '');



 %Getting BASE_NAME_2
 image_name=[FULL_FOLDER_NAME,'\',BASE_NAME1,num2str(MIN),'*',num2str(1),'.TIF'];
 TEMP3=dir(image_name);
 BASE_NAME2=TEMP3.name;
 string_to_remove1_1=[BASE_NAME1,num2str(MIN),num2str(0),num2str(0)];
 string_to_remove1_2=[BASE_NAME1,num2str(MIN),num2str(0)];
 string_to_remove1_3=[BASE_NAME1,num2str(MIN)];
 string_to_remove2=[num2str(1),'.TIF'];
 BASE_NAME2=strrep(BASE_NAME2,string_to_remove1_1, '');
 BASE_NAME2=strrep(BASE_NAME2,string_to_remove1_2, '');
 BASE_NAME2=strrep(BASE_NAME2,string_to_remove1_3, '');
 BASE_NAME2=strrep(BASE_NAME2,string_to_remove2, '');
 BASE_NAME2=strrep(BASE_NAME2,'_thumb', '');
 TIME=TEMP3.date;


 %Checking the maximum
 MAX=max(ACQUISITION_NUMBER);
```

```
image_name=[FULL_FOLDER_NAME,'\',BASE_NAME1,num2str(MAX),BASE_NAME2,num2str(Number_of_ima
ge_per_acquisition),'.TIF'];

         A=exist(image_name,'file');
         if A==0
             MAX=MAX-1;
         else
             MAX=MAX;
         end


         Folder_TOTAL=[Folder_TOTAL cellstr(FULL_FOLDER_NAME)];
         BASE_NAME1_TOTAL=[BASE_NAME1_TOTAL cellstr(BASE_NAME1)];
         BASE_NAME2_TOTAL=[BASE_NAME2_TOTAL cellstr(BASE_NAME2)];
         TIME_TOTAL=[TIME_TOTAL  cellstr(TIME)];

         MIN_TOTAL=[MIN_TOTAL;MIN];
         MAX_TOTAL=[MAX_TOTAL;MAX];


STRUCTURE=struct('EXPERIMENT_NAME',Experience_name,'FOLDER',FULL_FOLDER_NAME,'DATE',TIME,
'BASE_NAME1',BASE_NAME1,'BASE_NAME2',BASE_NAME2,'MIN',MIN,'MAX',MAX,'DATE_NUM',datenum(TI
ME),'Number_of_wells',Number_of_wells,'Number_of_acquisition_per_well',Number_of_image_pe
r_acquisition,'Passage',Passage,'Medium',MEDIUM_OUT);


         DATA=[DATA STRUCTURE];

    end
end

DATA_OUT=struct2cell(DATA);
DATA_OUT=DATA_OUT(:,:);
DATA_OUT=sortrows(DATA_OUT',8);


OUTPUT_NAME=[Experience_name,'_','DATA_STRUCTURE'];

save(OUTPUT_NAME,'DATA_OUT');

%load('DATA_STRUCTURE.mat','DATA_OUT')
```

METAMORPH_IMAGE_PROCESSING_v0r11.m

```
%M-file for processing images from Metamorph
%PM Juneau, August 2010

%v0r1: initial version
%v0r2: complete analysis of a whole well
%v0r3: modifications for 24 well plate analysis
%v0r4: Establish a model image for the first image of the serie used to
%calibrate segmentation model
%v0r5: including parameter for number of wells
%v0r6: incorporating a function for texture analysis
%v0r7: calculating the confidence interval for the parameters for the image
%segmentation function
%v0r8: giving more options for path names
%v0r9: incorporating a size distribution indicator
%v0r10: saving processed images
%v0r11: using datastructure for each experiment

%%INPUT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all

load('Experiment_April_08_2011_DATA_STRUCTURE');

filter_size=11;
Thresh=60;
cut_cells_regions=1000;

print_image=0; %If 1, we print the image

Total_number_of_scans=1000;
Max_number_of_variables=200;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Getting data from datastructure

NAME_OF_EXPERIMENT=DATA_OUT(1,1);
NUMBER_OF_WELLS=cell2mat(DATA_OUT(1,9));
Images_per_well=cell2mat(DATA_OUT(1,10));
Folder=DATA_OUT(:,2)';
Max=cell2mat(DATA_OUT(:,7)');
Min=cell2mat(DATA_OUT(:,6)');
Base_name1=DATA_OUT(:,4)';
Base_name2=DATA_OUT(:,5)';



%% Image processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nnn=1;

for h=1:NUMBER_OF_WELLS

    Cumul_dates=[];
    VARIABLES=[];
    Data=zeros(Images_per_well,Total_number_of_scans,200);


    for Image_number=1:Images_per_well
        [a,b]=size(Folder);
        ACQUISITION_NUMBER=1;

        for i=1:b

            begin=Min(i)+h-1;

            index=begin;
```

```
            while index<=Max(i)

                if index==0


image_name=[char(Folder(i)),'\',char(Base_name1(i)),'',char(Base_name2(i)),num2str(Image_
number),'.tif'];
                else

image_name=[char(Folder(i)),'\',char(Base_name1(i)),num2str(index),char(Base_name2(i)),nu
m2str(Image_number),'.tif'];
                end


                IMAGE=imread(image_name);

                if Image_number==1

                    Carac=struct2cell(dir(image_name));

                    Date_time=Carac(2);

                    Cumul_dates=[Cumul_dates Carac(2)];

                end

                    VARIABLES=[];
                    %%
                    %Segmentation of cell clusters


[I_out1,I_bin_regions,AVERAGE_BACKGROUND,STD_BACKGROUND,DATA_OUTPUT,VARIABLES_NAMES]=Meta
morph_CELL_CLUSTERS_SEGMENTATION_v0r2(IMAGE,filter_size,Thresh,cut_cells_regions);
                    [e,f]=size(DATA_OUTPUT);
                    Data(Image_number,ACQUISITION_NUMBER,1:f)=DATA_OUTPUT;
                    VARIABLES=[VARIABLES VARIABLES_NAMES];



                    %%
                    %Texture analysis of the segmented image



                    %%
                    %Size analysis of the segmented image


                    %%

                    if print_image==1

                        IMAGE_TO_SAVE=label2rgb(uint8(IMAGE),gray(256));

                        [r,s]=size(IMAGE);

                        for f=1:r
                            for g=1:s
                                if I_bin_regions(f,g)>0
                                    IMAGE_TO_SAVE(f,g,1)=255;
                                end
                            end
                        end

                        if index==0


save_name=['Trial',num2str(i),'_',char(Base_name1(i)),'',char(Base_name2(i)),num2str(Imag
e_number),'_PROCESSED','.jpg'];

                        else
```

```
save_name=['Trial',num2str(i),'_',char(Base_name1(i)),num2str(index),char(Base_name2(i)),
num2str(Image_number),'_PROCESSED','.jpg'];

                        end

                        imwrite(uint8(IMAGE_TO_SAVE),save_name,'jpg');
                        nnn=nnn+1


                    end

                    index=index+NUMBER_OF_WELLS;
                    ACQUISITION_NUMBER=ACQUISITION_NUMBER+1;

                end


            end

        end

        Savefile=[char(NAME_OF_EXPERIMENT),'_','Data_well_',num2str(h)];
        Savefile2=[char(NAME_OF_EXPERIMENT),'_','Time_well_',num2str(h)];
        Savefile3=[char(NAME_OF_EXPERIMENT),'_','VARIABLES'];

        save(Savefile,'Data');
        save(Savefile2,'Cumul_dates');
        save(Savefile3,'VARIABLES');

    end
```

Metamorph_DATAVIEWER_v0r6.m

```
%% Data
% This function aims at analyzing the data from Metamorph Image processing
% techniques


%%
%Kinetic models to fit...
function Metamorph_DATAVIEWER_v0r6(NAME_DATA_STRUCTURE,VECTOR_INPUT)


Monod_model_to_use=VECTOR_INPUT(1);
Deasy_Jankowski_model_to_use=VECTOR_INPUT(2);
Logistic_model_to_use=VECTOR_INPUT(3);
Richards_model_to_use=VECTOR_INPUT(4);
Gompertz_model_to_use=VECTOR_INPUT(5);
Weibull_model_to_use=VECTOR_INPUT(6);
H1_model_to_use=VECTOR_INPUT(7);
H2_model_to_use=VECTOR_INPUT(8);
H3_model_to_use=VECTOR_INPUT(9);
Michaelis_Menten_model_to_use=VECTOR_INPUT(10);
MIN_POINTS=VECTOR_INPUT(11);
MAX_POINTS=VECTOR_INPUT(12);
MAX_POINT=VECTOR_INPUT(13);



load(NAME_DATA_STRUCTURE);
Position_of_surface_in_data=3;


NAME_OF_EXPERIMENT=char(DATA_OUT(1,1));
NUMBER_OF_WELLS=cell2mat(DATA_OUT(1,9));
PASSAGE=DATA_OUT(1,11);
PASSAGE=PASSAGE{1,1};
MEDIUM=DATA_OUT(1,12);
MEDIUM=MEDIUM{1,1};
Cell_concentration=DATA_OUT(1,13);
Cell_concentration=Cell_concentration{1,1};
Cell_type=DATA_OUT(1,14);
Cell_type=Cell_type{1,1};
NAME_IMAGE_QUALITY=[NAME_OF_EXPERIMENT,'_IMAGE_QUALITY.xlsx'];


%Fit Monod model? =1 if yes

%Monod_model_to_use=1;
%MIN_POINTS=5;
MIN_POINTS_Deasy_Jankowski=MIN_POINTS;
%MAX_POINTS=50;
MAX_POINTS_Deasy_Jankowski=MAX_POINTS;
Enable_screening=0;
Thresh_screening=0.03;


%Fit Logistic model? =1 if yes

%MAX_POINT=60;

%Logistic_model_to_use=1;
%Richards_model_to_use=1;
%Gompertz_model_to_use=1;
%Weibull_model_to_use=1;
%Deasy_Jankowski_model_to_use=1;
%H1_model_to_use=1;
%H2_model_to_use=1;
%H3_model_to_use=1;
%Michaelis_Menten_model_to_use=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%


OUT=[];
MEDIUM_VECTOR=[];
CELL_TYPE_VECTOR=[];
IMAGE_QUALITY_VECTOR=[];

for ii=1:NUMBER_OF_WELLS


DATA_NAME=[NAME_OF_EXPERIMENT,'_','Data_well_',num2str(ii)];
DATE_NAME=[NAME_OF_EXPERIMENT,'_','Time_well_',num2str(ii)];
VARIABLES_NAMES=[NAME_OF_EXPERIMENT,'_','VARIABLES'];

Data=cell2mat(struct2cell(load(DATA_NAME)));
Date=load(DATE_NAME);
DATE_a=datenum(Date.Cumul_dates);
[k,l]=size(DATE_a);
[o,p,q]=size(Data);
DATE_NUM=zeros(k,1);

VARIABLES=load(VARIABLES_NAMES);


for i=1:k
    DATE_NUM(i,1)=(DATE_a(i,1)-DATE_a(1,1))*24;
end

%%
%Get portion of surface data

Portion_of_surface=mean(Data(:,1:k,Position_of_surface_in_data))';

%figure(1),
%plot(DATE_NUM,Portion_of_surface);





%%
Monod_parameters=zeros(1,6);
Time_opt=0;
Y_opt=0;
VARIABLES_MONOD={'Exponential  Mhum(h-1)'  'Interval'  'Exponential  X0'  'Interval'  'R2
Exponential' 'Number points Exponential'};
if Monod_model_to_use==1

    n_points=MIN_POINTS;

    R2_opt=0;
    p_est_opt=zeros(1,2);
    INT_PARAM_opt=zeros(1,2);
    N_points_opt=0;

    if k>=MIN_POINTS;

    while n_points<=MAX_POINTS && n_points<=k
    Yexp=Portion_of_surface(1:n_points);

    Time=round(DATE_NUM(1:n_points));

    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
```

```
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.05, 0.1];

    options = statset('MaxIter',10000,'FunValCheck','off');
    Time;
    Yexp;
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@Monod_model,p_init,options);
    Yest=Monod_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);

    R2=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2))];   %   intervalles   de   confiance
individuels à 95%

    %** Affiche les résultats à l'écran *****
    %disp(' Parameters:')
    %disp([' Mhum (h-1) = ', num2str(p_est(1,1)), ' +/- ', num2str(INT_PARAM(1,1))])
    %disp([' X0 = ', num2str(p_est(1,2)), ' +/- ', num2str(INT_PARAM(2,1))])

    if R2>R2_opt
        p_est_opt=p_est;
        INT_PARAM_opt=INT_PARAM;
        R2_opt=R2;
        N_points_opt= n_points;
        Time_opt=Time;
        Y_opt=Yest;
    end

    figure(1),
    plot(Time,Yexp,Time,Yest);

     n_points=n_points+1;

    end
    end

    Monod_parameters=[p_est_opt(1,1)  INT_PARAM_opt(1,1)  p_est_opt(1,2)  INT_PARAM_opt(2,1)
R2_opt N_points_opt];

    figure(2),
    subplot(3,3,1),scatter(Time,Yexp); hold on
    plot(Time_opt,Y_opt,'r'), title('Exponential Model');
    hold off

    figure(11),
    scatter(Time,Yexp); hold on
    plot(Time_opt,Y_opt,'r'),                                          title('Exponential
Model'),xlabel('Time(h)'),ylabel('Surface Recovered (-)'),ylim([0 1]);
    hold off

    MHUM_EXPONENTIAL=p_est_opt(1,1);
    MHUM_EXPONENTIAL_INTERVAL=INT_PARAM_opt(1,1);
```

```
    X_0_EXPONENTIAL=p_est_opt(1,2);
    X_0_EXPONENTIAL_INTERVAL=INT_PARAM_opt(2,1);


    save('MHUM_EXPONENTIAL','MHUM_EXPONENTIAL');
    save('X_0_EXPONENTIAL','X_0_EXPONENTIAL');

end


%%
Deasy_Jankowski_parameters=zeros(1,8);
Time_opt=0;
Y_opt=0;
VARIABLES_Deasy_Jankowski={'Deasy_Jankowski   Alpha'   'Interval'   'Deasy_Jankowski   DT'
'Interval'   'Deasy_Jankowski   X_O'   'Interval'   'R2   Deasy_Jankowski'   'Number   points
Deasy_Jankowski'};

if Deasy_Jankowski_model_to_use==1

    n_points=MIN_POINTS_Deasy_Jankowski;

    R2_opt=0;
    p_est_opt=zeros(1,2);
    INT_PARAM_opt=zeros(1,2);
    N_points_opt=0;

    if k>=MIN_POINTS_Deasy_Jankowski;

    while n_points<=MAX_POINTS_Deasy_Jankowski && n_points<=k

    Yexp=Portion_of_surface(1:n_points);

    Time=round(DATE_NUM(1:n_points));

    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.1 24 0.05];


    %,'DerivStep',[1e-3 1e-1 1e-3]

    %options = statset('MaxIter',10000,'FunValCheck','off');
    %[p_est,RESIDUAL,Jacobian]                                                                =
nlinfit(Time,Yexp,@Deasy_Jankowski_model,p_init,options);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%For fmincon
    save('BUFFER_DEASY_YEXP','Yexp');
    save('BUFFER_DEASY_TIME','Time');

    options=optimset('MaxIter',500,'Algorithm','interior-point');
    p_est=fmincon(@Deasy_Jankowski_model_FMINCON,p_init,[],[],[],[],[0    0    0],[1    100
1],[],options);
```

```
    Yest=Deasy_Jankowski_model(p_est,Time);

    RESIDUAL=Yexp-Yest;

    delta=1e-3;

    Jacobian=[(Deasy_Jankowski_model((p_est+[delta              0              0]),Time)-
Deasy_Jankowski_model((p_est-[delta                                                  0
0]),Time))./(2*delta),(Deasy_Jankowski_model((p_est+[0          delta          0]),Time)-
Deasy_Jankowski_model((p_est-[0                                                  delta
0]),Time))./(2*delta),(Deasy_Jankowski_model((p_est+[0          0          delta]),Time)-
Deasy_Jankowski_model((p_est-[0 0 delta]),Time))./(2*delta)];

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    Yest=Deasy_Jankowski_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);

    R2_Deasy_Jankowski=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3))]; % intervalles de confiance individuels à 95%

    %** Affiche les résultats à l'écran *****
    %disp(' Parameters:')
    %disp([' Mhum (h-1) = ', num2str(p_est(1,1)), ' +/- ', num2str(INT_PARAM(1,1))])
    %disp([' X0 = ', num2str(p_est(1,2)), ' +/- ', num2str(INT_PARAM(2,1))])

    if R2_Deasy_Jankowski>R2_opt
        p_est_opt=p_est;
        INT_PARAM_opt=INT_PARAM;
        R2_opt=R2_Deasy_Jankowski;
        N_points_opt= n_points;
        Time_opt=Time;
        Y_opt=Yest;
    end

    figure(3),
    plot(Time,Yexp,Time,Yest);

     n_points=n_points+1;

    end
    end

    Deasy_Jankowski_parameters=[p_est(1,1)   INT_PARAM(1,1)   p_est(1,2)   INT_PARAM(2,1)
p_est(1,3) INT_PARAM(3,1) R2_Deasy_Jankowski N_points_opt];

    figure(2),
    subplot(3,3,2),scatter(Time,Yexp); hold on
    plot(Time_opt,Y_opt,'r'),          title('Deasy           and            Jankowski
Model'),xlabel('Time(h)'),ylabel('Surface Recovered (-)');
    hold off

    figure(12),
    scatter(Time,Yexp); hold on
    plot(Time_opt,Y_opt,'r'),   title('Sherley   Model'),xlabel('Time(h)'),ylabel('Surface
Recovered (-)'),ylim([0 1]);
    hold off


end
```

```
%%
Logistic_parameters=zeros(1,8);
VARIABLES_Logistic={'Logistic Cmax(h-1)' 'Interval' 'Logistic Xmax' 'Interval' 'Logistic
X0' 'Interval' 'R2 logistic' 'Number points logistic'};

if Logistic_model_to_use==1

     Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end



    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.05, 0.95 0];

    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@Logistic_model,p_init,options);
    Yest=Logistic_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_logistic=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3))]; % intervalles de confiance individuels à 95%

    %** Affiche les résultats à l'écran *****
    %disp(' Parameters:')
    %disp([' Cmax (h-1) = ', num2str(p_est(1,1)), ' +/- ', num2str(INT_PARAM(1,1))])
    %disp([' Xmax = ', num2str(p_est(1,2)), ' +/- ', num2str(INT_PARAM(2,1))])
    %disp([' X0 = ', num2str(p_est(1,3)), ' +/- ', num2str(INT_PARAM(3,1))])

    %figure(5),
    %plot(Time,Yexp,Time,Yest);

    Logistic_parameters=[p_est(1,1)  INT_PARAM(1,1)  p_est(1,2)  INT_PARAM(2,1)  p_est(1,3)
INT_PARAM(3,1)  R2_logistic k];
```

```
    figure(2),
    subplot(3,3,3),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('Logistic Model')
    hold off

    figure(13),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'),     title('Logistic     Model'),xlabel('Time(h)'),ylabel('Surface
Recovered (-)'),ylim([0 1]);
    hold off

end

%%
Richards_parameters=zeros(1,10);
VARIABLES_Richards={'Richards Beta' 'Interval' 'Richards Gamma' 'Interval' 'Richards M'
'Interval' 'Richards X_O' 'Interval' 'R2 Richards' 'Number points Richards'};

if Richards_model_to_use==1

      Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end


    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.1 0.01 0.5 0.1];

    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@Richards_model,p_init,options);
    Yest=Richards_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_Richards=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
```

```
p)*sqrt(VAR_PARAM(3,3));tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(4,4))];   %
intervalles de confiance individuels à 95%


    %figure(6),
    %plot(Time,Yexp,Time,Yest);

    Richards_parameters=[p_est(1,1)  INT_PARAM(1,1)  p_est(1,2)  INT_PARAM(2,1)  p_est(1,3)
INT_PARAM(3,1) p_est(1,4) INT_PARAM(4,1)  R2_Richards k];

    figure(2),
    subplot(3,3,4),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('Richards Model')
    hold off

    figure(14),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'),     title('Richards     Model'),xlabel('Time(h)'),ylabel('Surface
Recovered (-)'),ylim([0 1]);
    hold off

end

%%
Gompertz_parameters=zeros(1,8);
VARIABLES_Gompertz={'Gompertz  Beta'  'Interval'  'Gompertz  M'  'Interval'  'Gompertz  X_O'
'Interval' 'R2 Gompertz' 'Number points Gompertz'};

if Gompertz_model_to_use==1

    Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end

    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.1 0.5 0.1];

    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@Gompertz_model,p_init,options);
    Yest=Gompertz_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
```

```
    R2_Gompertz=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3))]; % intervalles de confiance individuels à 95%


    %figure(6),
    %plot(Time,Yexp,Time,Yest);

    Gompertz_parameters=[p_est(1,1)  INT_PARAM(1,1)  p_est(1,2)  INT_PARAM(2,1)  p_est(1,3)
INT_PARAM(3,1) R2_Gompertz k];

    figure(2),
    subplot(3,3,5),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('Gompertz Model')
    hold off

    figure(15),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'),     title('Gompertz     Model'),xlabel('Time(h)'),ylabel('Surface
Recovered (-)'),ylim([0 1]);
    hold off

end

%%
Weibull_parameters=zeros(1,10);
VARIABLES_Weibull={'Weibull  Beta'  'Interval'  'Weibull  Gamma'  'Interval'  'Weibull  M'
'Interval' 'Weibull X_O' 'Interval' 'R2 Weibull' 'Number points Weibull'};

if Weibull_model_to_use==1

     Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end


    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[0.1 0.01 0.5 0.1];
```

```
    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@Weibull_model,p_init,options);
    Yest=Weibull_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_Weibull=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3));tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(4,4))];   %
intervalles de confiance individuels à 95%


    %figure(6),
    %plot(Time,Yexp,Time,Yest);

    Weibull_parameters=[p_est(1,1)  INT_PARAM(1,1)  p_est(1,2)  INT_PARAM(2,1)  p_est(1,3)
INT_PARAM(3,1) p_est(1,4) INT_PARAM(4,1)  R2_Weibull k];

    figure(2),
    subplot(3,3,6),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('Weibull Model')
    hold off

end

%%
H1_parameters=zeros(1,10);
VARIABLES_H1={'H1  Beta'  'Interval'  'H1  Theta'  'Interval'  'H1  M'  'Interval'  'H1  X_O'
'Interval' 'R2 H1' 'Number points H1'};

if H1_model_to_use==1

      Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end


    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end
```

```
    p_init=[0 0.01 0.5 0];

    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@H1_model,p_init,options);
    Yest=H1_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_H1=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3));tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(4,4))];   %
intervalles de confiance individuels à 95%


    %figure(6),
    %plot(Time,Yexp,Time,Yest);

    H1_parameters=[p_est(1,1)   INT_PARAM(1,1)   p_est(1,2)   INT_PARAM(2,1)   p_est(1,3)
INT_PARAM(3,1) p_est(1,4) INT_PARAM(4,1)  R2_H1 k];

    figure(2),
    subplot(3,3,7),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('H1 Model')
    hold off

    figure(16),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('H1 Model'),xlabel('Time(h)'),ylabel('Surface Recovered (-
)'),ylim([0 1]);
    hold off

end

%%
H2_parameters=zeros(1,12);
VARIABLES_H2={'H2 Alpha' 'Interval' 'H2 Beta' 'Interval' 'H2 Gamma' 'Interval' 'H2 M'
'Interval' 'H2 X_O' 'Interval' 'R2 H2' 'Number points H2'};

if H2_model_to_use==1

     Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end


    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
```

```
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[10 1e-7 3 0.9 0.05];

    options = statset('MaxIter',10000,'FunValCheck','off','DerivStep',[1e-1 1e-8 1e-1 1e-
1 1e-1]);
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@H2_model,p_init,options);
    Yest=H2_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_H2=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(4,4));tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(5,5))];   %
intervalles de confiance individuels à 95%


    %figure(7),
    %plot(Time,Yexp,Time,Yest);

    H2_parameters=[p_est(1,1)   INT_PARAM(1,1)   p_est(1,2)   INT_PARAM(2,1)   p_est(1,3)
INT_PARAM(3,1) p_est(1,4) INT_PARAM(4,1) p_est(1,5) INT_PARAM(5,1) R2_H2 k];

    figure(2),
    subplot(3,3,8),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('H2 Model')
    hold off

end

%%
H3_parameters=zeros(1,12);
VARIABLES_H3={'H3 L' 'Interval' 'H3 Delta' 'Interval' 'H3 Gamma' 'Interval' 'H3 Theta'
'Interval' 'H3 X_O' 'Interval' 'R2 H3' 'Number points H3'};

if H3_model_to_use==1

      Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end


    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
```

```
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[1.2 0.4 (0.00001) 0.01 0.1];

    options = statset('MaxIter',10000,'FunValCheck','off','Robust','on','DerivStep',[1e-1
1e-2 1e-15 1e-10 1e-1]);
    [p_est,RESIDUAL,Jacobian] = nlinfit(Time,Yexp,@H3_model,p_init,options);

%[p_est,fval,exitflag,output,lambda,grad,hessian]=fmincon(H3_model_to_minimize,p_init,-
1.*ones(1,5),zeros(1,5));
  %[1e-15 1e-15 1e-15 1e-15 1e-15]

    Yest=H3_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_H3=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(3,3));tinv(((1-Test_level)/2)+Test_level,n-
p)*sqrt(VAR_PARAM(4,4));tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(5,5))];   %
intervalles de confiance individuels à 95%


    %figure(8),
    %plot(Time,Yexp,Time,Yest);

    H3_parameters=[p_est(1,1)   INT_PARAM(1,1)   p_est(1,2)   INT_PARAM(2,1)   p_est(1,3)
INT_PARAM(3,1) p_est(1,4) INT_PARAM(4,1) p_est(1,5) INT_PARAM(5,1) R2_H3 k];

    figure(2),
    subplot(3,3,9),scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('H3 Model')
    hold off

end

%%
Michaelis_Menten_parameters=zeros(1,10);
VARIABLES_Michaelis_Menten={'MM  Mhum'  'Interval'  'MM  X_M'  'Interval'  'MM  Ks_Y_xs'
'Interval' 'MM X_O' 'Interval' 'R2 MM' 'Number points MM'};

if Michaelis_Menten_model_to_use==1 && Monod_model_to_use==1

      Time=round(DATE_NUM);

    if k>MAX_POINT

        Yexp=Portion_of_surface(1:MAX_POINT,1);

        Time=Time(1:MAX_POINT,1);

    else
        Yexp=Portion_of_surface;

    end
```

```
    if Enable_screening==1

        Inter_time=[];
        Inter_yexp=[];
        [aa,bb]=size(Time);

        for d=1:aa
            if Yexp(d)>=Thresh_screening
                Inter_time=[Inter_time;Time(d)];
                Inter_yexp=[Inter_yexp;Yexp(d)];
            end
        end

        Time=Inter_time;
        Yexp=Inter_yexp;

    end


    p_init=[1 0.5];

    options = statset('MaxIter',10000,'FunValCheck','off');
    [p_est,RESIDUAL,Jacobian]                                                                    =
nlinfit(Time,Yexp,@Michaelis_Menten_model,p_init,options);
    Yest=Michaelis_Menten_model(p_est,Time);

    n=size(Time,1);
    p=size(p_est,1);
    MSE=sum(RESIDUAL.^2,1)/(n-p);
    R2_Michaelis_Menten=1-sum(RESIDUAL.^2,1)/sum((Yexp-mean(Yexp)).^2);

    VAR_PARAM=inv(Jacobian'*Jacobian)*MSE;   %   variance   des   estimés   (approximation
linéaire)
    Test_level=0.95;
    INT_PARAM=[tinv(((1-Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(1,1));tinv(((1-
Test_level)/2)+Test_level,n-p)*sqrt(VAR_PARAM(2,2))];   %   intervalles   de   confiance
individuels à 95%


    %figure(6),
    %plot(Time,Yexp,Time,Yest);

    Michaelis_Menten_parameters=[MHUM_EXPONENTIAL   MHUM_EXPONENTIAL_INTERVAL   p_est(1,1)
INT_PARAM(1,1)   p_est(1,2)   INT_PARAM(2,1)   X_0_EXPONENTIAL   X_0_EXPONENTIAL_INTERVAL
R2_Michaelis_Menten k];

    figure(3),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'), title('Michaelis Menten Model')
    hold off

    figure(17),
    scatter(Time,Yexp); hold on
    plot(Time,Yest,'r'),                      title('Michaelis                      Menten
Model'),xlabel('Time(h)'),ylabel('Surface Recovered (-)'),ylim([0 1]);
    hold off

end

%%
%Get the data out

INTER=struct2cell(VARIABLES);

HEADLINE=(['Experience ID' 'IMAGE QUALITY' 'Cells' 'Medium' 'Well' 'Passage' 'Cells
Initial Concentration' 'Time' INTER{1,1} VARIABLES_MONOD VARIABLES_Deasy_Jankowski
VARIABLES_Logistic VARIABLES_Richards VARIABLES_Gompertz VARIABLES_Weibull VARIABLES_H1
VARIABLES_H2 VARIABLES_H3 VARIABLES_Michaelis_Menten]);
```

```
[a,b]=size(INTER{1,1});

DATA_OUTPUT=[];

[num,IMAGE_QUALITY_STRINGS,raw]=xlsread(NAME_IMAGE_QUALITY,num2str(ii));

IMAGE_QUALITY_VECTOR=[IMAGE_QUALITY_VECTOR;IMAGE_QUALITY_STRINGS];

Well_vector=ones(k,1).*ii;

Passage_vector=ones(k,1).*PASSAGE(ii);

Cells_conc_vector=ones(k,1).*Cell_concentration(ii);

Monod_vector=ones(k,1)*Monod_parameters;

Deasy_Jankowski_vector=ones(k,1)*Deasy_Jankowski_parameters;

Logistic_vector=ones(k,1)*Logistic_parameters;

Richards_vector=ones(k,1)*Richards_parameters;

Gompertz_vector=ones(k,1)*Gompertz_parameters;

Weibull_vector=ones(k,1)*Weibull_parameters;

H1_vector=ones(k,1)*H1_parameters;

H2_vector=ones(k,1)*H2_parameters;

H3_vector=ones(k,1)*H3_parameters;

Michaelis_Menten_vector=ones(k,1)*Michaelis_Menten_parameters;

DATA_OUTPUT=[Well_vector Passage_vector Cells_conc_vector DATE_NUM];


for jj=1:k

    MEDIUM_VECTOR=[MEDIUM_VECTOR;cellstr(MEDIUM(ii))];
    CELL_TYPE_VECTOR=[CELL_TYPE_VECTOR;cellstr(Cell_type(ii))];

end


for j=1:b

DATA_OUTPUT=[DATA_OUTPUT mean(Data(:,1:k,j))'];

end

DATA_OUTPUT=[DATA_OUTPUT      Monod_vector      Deasy_Jankowski_vector      Logistic_vector
Richards_vector    Gompertz_vector    Weibull_vector    H1_vector    H2_vector    H3_vector
Michaelis_Menten_vector];


OUT=[OUT;DATA_OUTPUT];


Name_Figure=[NAME_OF_EXPERIMENT,'_Well_',num2str(ii)];

Name_Figure2=[NAME_OF_EXPERIMENT,'_MICHAELIS_MENTEN','_Well_',num2str(ii)];

saveas(figure(2),Name_Figure,'jpg')

saveas(figure(3),Name_Figure2,'jpg')


end
```

```
filename=[NAME_OF_EXPERIMENT,'_','DATA'];
xlswrite(filename,HEADLINE,1,'A1');

EXPERIMENT_ID=[];

[p,l]=size(OUT);

for v=1:p

    EXPERIMENT_ID=[EXPERIMENT_ID;cellstr(NAME_OF_EXPERIMENT)];


end

xlswrite(filename,EXPERIMENT_ID,1,'A2');

xlswrite(filename,IMAGE_QUALITY_VECTOR,1,'B2');

xlswrite(filename,CELL_TYPE_VECTOR,1,'C2');

xlswrite(filename,MEDIUM_VECTOR,1,'D2');

xlswrite(filename,OUT,1,'E2');
```

Monod_model.m

```
function Yest = Monod_model(p_var,T)

mhum=p_var(1,1); %paramètre mhumax
Y_O=p_var(1,2);


Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;

    k1=Y_1*mhum;

    k2=(Y_1+0.5*h*k1)*mhum;

    k3=(Y_1+0.5*h*k2)*mhum;

    k4=(Y_1+h*k3)*mhum;

    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Logistic_model.m

```
function Yest = Logistic_model(p_var,T)

Cmax=p_var(1,1);
Xmax=p_var(1,2);
X_O=p_var(1,3);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;

    k1=Y_1*Cmax*(1-Y_1/Xmax);

    k2=(Y_1+0.5*h*k1)*Cmax*(1-(Y_1+0.5*h*k1)/Xmax);

    k3=(Y_1+0.5*h*k2)*Cmax*(1-(Y_1+0.5*h*k2)/Xmax);

    k4=(Y_1+h*k3)*Cmax*(1-(Y_1+h*k3)/Xmax);

    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Gompertz_model.m

```
function Yest = Gompertz_model(p_var,T)

Beta=p_var(1,1);
M=p_var(1,2);
X_O=p_var(1,3);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    ALPHA=log(M/Y_O);

    Y(i+1)=M*exp(-ALPHA*exp(-M*Beta*t(i)));
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Richards_model.m

```
function Yest = Richards_model(p_var,T)

Beta=p_var(1,1);
Gamma=p_var(1,2);
M=p_var(1,3);
X_O=p_var(1,4);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    ALPHA=(M/Y_O)^(1/Gamma)-1;

    Y(i+1)=M/((1+ALPHA*exp(-M*Beta*t(i)))^Gamma);
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Weibull_model.m

```
function Yest = Weibull_model(p_var,T)

Beta=p_var(1,1);
Gamma=p_var(1,2);
M=p_var(1,3);
X_O=p_var(1,4);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    ALPHA=M-Y_O;

    Y(i+1)=M-ALPHA*exp(-Beta*t(i)*Gamma);
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

H1_model.m

```
function Yest = H1_model(p_var,T)

Beta=p_var(1,1); %paramètre mhumax
Theta=p_var(1,2);
M=p_var(1,3);
X_O=p_var(1,4);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;

    k1=Y_1/M*(M-Y_1)*(M*Beta+Theta/sqrt(1+t(i)^2));

    k2=(Y_1+0.5*h*k1)/M*(M-(Y_1+0.5*h*k1))*(M*Beta+Theta/sqrt(1+(t(i)+h/2)^2));

    k3=(Y_1+0.5*h*k2)/M*(M-(Y_1+0.5*h*k2))*(M*Beta+Theta/sqrt(1+(t(i)+h/2)^2));

    k4=(Y_1+h*k3)/M*(M-(Y_1+h*k3))*(M*Beta+Theta/sqrt(1+(t(i)+h)^2));

    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

H2_model.m

```
function Yest = H2_model(p_var,T)

Alpha=p_var(1,1); %paramètre mhumax
Beta=p_var(1,2);
Gamma=p_var(1,3);
M=p_var(1,4);
X_O=p_var(1,5);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;

    k1=Alpha*Beta*Gamma*(Y_1)^2*(t(i))^(Gamma-1)*tanh((M-(Y_1))/(Alpha*(Y_1)));

    k2=Alpha*Beta*Gamma*(Y_1+0.5*h*k1)^2*(t(i)+h/2)^(Gamma-1)*tanh((M-
(Y_1+0.5*h*k1))/(Alpha*(Y_1+0.5*h*k1)));

    k3=Alpha*Beta*Gamma*(Y_1+0.5*h*k2)^2*(t(i)+h/2)^(Gamma-1)*tanh((M-
(Y_1+0.5*h*k2))/(Alpha*(Y_1+0.5*h*k2)));

    k4=Alpha*Beta*Gamma*(Y_1+h*k3)^2*(t(i)+h)^(Gamma-1)*tanh((M-
(Y_1+h*k3))/(Alpha*(Y_1+h*k3)));

    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

H3_model.m

```
function Yest = H3_model(p_var,T)

L=p_var(1,1); %paramètre mhumax
Delta=p_var(1,2);
Gamma=p_var(1,3);
Theta=p_var(1,4);
X_O=p_var(1,5);

Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;

    %Delta=(L-(Y_1))*(Delta*Gamma*((t(i))^(Gamma-1))+Theta/sqrt(1+Theta^2*(t(i))^2));

    k1=(L-(Y_1))*(Delta*Gamma*(t(i))^(Gamma-1)+Theta/sqrt(1+Theta^2*(t(i))^2));

    k2=(L-(Y_1+0.5*h*k1))*(Delta*Gamma*(t(i)+h/2)^(Gamma-
1)+Theta/sqrt(1+Theta^2*(t(i)+h/2)^2));

    k3=(L-(Y_1+0.5*h*k2))*(Delta*Gamma*(t(i)+h/2)^(Gamma-
1)+Theta/sqrt(1+Theta^2*(t(i)+h/2)^2));

    k4=(L-(Y_1+h*k3))*(Delta*Gamma*(t(i)+h)^(Gamma-1)+Theta/sqrt(1+Theta^2*(t(i)+h)^2));

    if t(i)==0

        %Delta=0;

        k1=0;

        k2=0;

        k3=0;

        k4=0;

    end


    %Y_2=Y(i)+Delta*h;
    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end
```

```
for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Michaelis_Menten_model.m

```
function Yest = Michaelis_Menten_model(p_var,T)

load('MHUM_EXPONENTIAL','MHUM_EXPONENTIAL');
load('X_0_EXPONENTIAL','X_0_EXPONENTIAL');

Mhum=MHUM_EXPONENTIAL; %paramètre mhumax
X_m=p_var(1,1);
Ks_Y_xs=p_var(1,2);
X_O=X_0_EXPONENTIAL;


Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;

for i=1:N

    Y_1=Y_2;


    k1=Mhum*(X_m-Y_1)*Y_1/(Ks_Y_xs+(X_m-Y_1));

    k2=Mhum*(X_m-(Y_1+0.5*h*k1))*(Y_1+0.5*h*k1)/(Ks_Y_xs+(X_m-(Y_1+0.5*h*k1)));

    k3=Mhum*(X_m-(Y_1+0.5*h*k2))*(Y_1+0.5*h*k2)/(Ks_Y_xs+(X_m-(Y_1+0.5*h*k2)));

    k4=Mhum*(X_m-(Y_1+h*k3))*(Y_1+h*k3)/(Ks_Y_xs+(X_m-(Y_1+h*k3)));

    Y_2=Y(i)+(1/6)*h*(k1+2*k2+2*k3+k4);


    Y(i+1)=Y_2;
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Deasy_Jankowski_model.m

```
function Yest = Deasy_Jankowski_model(p_var,T)

Alpha=p_var(1,1);
DT=p_var(1,2);
X_O=p_var(1,3);

%Parameters must be bounded

if X_O<0
    X_O=abs(X_O);
end

if Alpha>1
    Alpha=1;
end


Y_O=X_O;

Ymodel=[];

h=0.25;

Y_2=Y_O;

N=1000/h;

Y=[];
t=[];

Y(1)=Y_O;
t(1)=0;


difference=1;



for i=1:N


    Y(i+1)=Y_O*(0.5+(1-(2*Alpha)^(t(i)/DT+1))/(2*(1-2*Alpha)));
    t(i+1)=t(i)+h;

end


for i=1:length(T)
    Yest(i)=Y(find(t==T(i)));
end

Yest=Yest';
```

Deasy_Jankowski_model_FMINCON.m

```
function ERREUR = Deasy_Jankowski_model_FMINCON(p_var)


load('BUFFER_DEASY_YEXP','Yexp');
load('BUFFER_DEASY_TIME','Time');


Yest = Deasy_Jankowski_model(p_var,Time);



ERREUR=sum((Yest-Yexp).^2);
```

MIA_v0r9.m

```
%This function is for perfoming MIA on multiple RGB images. The data can be exported to
PROMV.

%Pierre-Marc Juneau, March 2014

clear
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LATENT_SPACE_GENERATION=0;
MINIMUM_MAXIMUM_SCORES_CALCULATION=0;
SCORE_DENSITY_HISTOGRAM_MATRIX=0;
INDIVIDUAL_SCORE_DENSITY_HISTOGRAM_MATRIX=0;
SHOW_SCORE_DENSITY_HISTOGRAM=0;
INDIVIDUAL_SHOW_SCORE_DENSITY_HISTOGRAM=0;
MASK_DATA_GENERATION=0;
MASK_FINDER_ROIPOLY=0;
X_MATRIX_GENERATION=0;
Y_MATRIX_GENERATION=0;
COVARIANCE_X_Y=0;
PHASE_ANGLE=0;
HISTOGRAMS_GENERATION=1;
PROMV_EXPORT=0;
CROSSVAL=0;
PLS_MODEL=0;

NUMBER_OF_COMPONENTS_PLS=10;

Y_MATRIX_SOURCE='D:\Analysis\4_Materna_Bunch_results_3\Y_Matrix_Materna.xlsx';


%%
MASK_NUMBER=1;

INVERSE_MASKING=0;

%%
MASK_SINGLE_IMAGE=0;
IMAGE_NUMBER_TO_ANALYZE=1;

Number_of_lambdas=3;

COMPOSANTE_NUM_1=1;
COMPOSANTE_NUM_2=2;


nbrgroupes=32;
%Paramenters for score density histogram display
NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM=10;
Decimation_factor=10;
HIGH_BOUND=0.5;


%Threshold=1;
IMAGE_FORMAT='.TIF';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%if SHOW_SCORE_DENSITY_HISTOGRAM~=1
%Get the directory where the images are
directory_name_image = uigetdir('','Choose the directory where the images are');
listing_image = dir([directory_name_image,'\','*.TIF']);
[Number_of_images,b]=size(listing_image);

IMAGES_NAME_LIST={listing_image.name}';


%end
```

```
%%
%Generating common latent space for all images

if LATENT_SPACE_GENERATION==1

    Z=zeros(Number_of_lambdas,Number_of_lambdas);
    save('Zmatrix.mat','Z');



    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;
        MATRIX_NAME=strrep(IMAGE_NAME,'.TIF', '');

        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);

        SPECTRAL=double(IMAGE);



        [a,b,c]=size(SPECTRAL);


        for i=1:b


            PARTIAL=reshape(SPECTRAL(:,i,:),a,c);


            Z=Z+PARTIAL'*PARTIAL;
            save('Zmatrix.mat','Z');

        end



    end

    save('Zmatrix.mat','Z');

end


%%
if MINIMUM_MAXIMUM_SCORES_CALCULATION==1

Z=load('Zmatrix.mat','Z');
Z=Z.Z;
[P,E,PT]=svd(Z);
%variancecaptured = sum(E)./sum(sum(E));
%variancecaptured1_10 = variancecaptured(1,1:10);

%%

%We find the minimum and the maximum value of all the scores

NUMBER_TOTAL_COMPONENTS=size(Z,1);

SCORES_MIN=ones(1,NUMBER_TOTAL_COMPONENTS).*realmax;
SCORES_MAX=ones(1,NUMBER_TOTAL_COMPONENTS).*(-realmax);

for m=1:Number_of_images
    m

    IMAGE_NAME=listing_image(m).name;

    IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);

    SPECTRAL=double(IMAGE);
```

```
    [a,b,c]=size(SPECTRAL);

    for i=1:b

        X =reshape(SPECTRAL(:,i,:),a,c)';
        T=X'*PT;

        for k=1:NUMBER_TOTAL_COMPONENTS
            SCORES_LOCAL_MAX=max(T);
            SCORES_LOCAL_MIN=min(T);

            if SCORES_LOCAL_MAX(k)>SCORES_MAX(k)
                SCORES_MAX(k)=SCORES_LOCAL_MAX(k);
            end

            if SCORES_LOCAL_MIN(k)<SCORES_MIN(k)
                SCORES_MIN(k)=SCORES_LOCAL_MIN(k);
            end

        end

    end

end

save('SCORES_MIN.mat','SCORES_MIN');
save('SCORES_MAX.mat','SCORES_MAX');

end

%%
%Computation of the density histogram matrix

if SCORE_DENSITY_HISTOGRAM_MATRIX==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);

    A = zeros(nbrgroupes,nbrgroupes);

    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);

    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;

        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);

        SPECTRAL=double(IMAGE);

        [a,b,c]=size(SPECTRAL);


        for i=1:b

            X =reshape(SPECTRAL(:,i,:),a,c)';
```

```
            t1=X'*PT(:,COMPOSANTE_NUM_1);
            t2=X'*PT(:,COMPOSANTE_NUM_2);


            s1  =  ones(size(t1,1),1)+round(((t1  -  (t1min*ones(size(t1,1),1)))./((t1max-
t1min))).*(nbrgroupes-1));
            s2  =  ones(size(t2,1),1)+round(((t2  -  (t2min*ones(size(t2,1),1)))./((t2max-
t2min))).*(nbrgroupes-1));

            for k = 1:length(s1)

                A(s2(k), s1(k)) = A(s2(k), s1(k)) + 1;

            end

        end
    end


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.
mat'];
    save(NAME,'A');


end


%%
if INDIVIDUAL_SCORE_DENSITY_HISTOGRAM_MATRIX==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);

    A = zeros(nbrgroupes,nbrgroupes);

    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);

    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;

        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);

        SPECTRAL=double(IMAGE);

        [a,b,c]=size(SPECTRAL);

        A = zeros(nbrgroupes,nbrgroupes);

        for i=1:b

            X =reshape(SPECTRAL(:,i,:),a,c)';
            t1=X'*PT(:,COMPOSANTE_NUM_1);
            t2=X'*PT(:,COMPOSANTE_NUM_2);


            s1  =  ones(size(t1,1),1)+round(((t1  -  (t1min*ones(size(t1,1),1)))./((t1max-
t1min))).*(nbrgroupes-1));
```

```
            s2  =  ones(size(t2,1),1)+round(((t2  -  (t2min*ones(size(t2,1),1)))./((t2max-
t2min))).*(nbrgroupes-1));

            for k = 1:length(s1)

                A(s2(k), s1(k)) = A(s2(k), s1(k)) + 1;

            end

        end


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
IMAGE_',IMAGE_NAME,'.mat'];
        save(NAME,'A');



    end

end



%%
%Showing the score density histogram

if SHOW_SCORE_DENSITY_HISTOGRAM==1


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.
mat'];
    A=load(NAME,'A');
    A=A.A;

    RELATIVE=max(max(A));

    SPLIT=256/NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM;

    [h,j] = find((round(RELATIVE*HIGH_BOUND))<=A);
    for m = 1:length(h),
        H(h(m),j(m)) = 256;
    end;

    for k=1:1:(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-3);
        k;

        [h,j]      =      find((round(RELATIVE*HIGH_BOUND/(Decimation_factor^k)))<=A      &
A<(round(RELATIVE*HIGH_BOUND/(Decimation_factor^(k-1)))));
        for m = 1:length(h),
            H(h(m),j(m))                                                                =
round(SPLIT*(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-k));  %  Those  bins  in  A
having 76-150 hits are reassigned to a pixel intensity value of 226
        end;


    end

    [h,j] = find(A==2);
    for m = 1:length(h),
        H(h(m),j(m)) = round(SPLIT*2);
    end;

    [h,j] = find(A==1);
    for m = 1:length(h),
        H(h(m),j(m)) = round(SPLIT*1);
    end;



    [h,j] = find(A==0);
```

```
    for m = 1:length(h),
        H(h(m),j(m)) = 0;
    end;


    colormap(hot(256));
    map = colormap;

    figure(1);
    imshow(H,map,'InitialMagnification','fit');%
    xlabel(num2str(COMPOSANTE_NUM_1));
    ylabel(num2str(COMPOSANTE_NUM_2));


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'.mat'];
    save(NAME,'H');


end

%%
if INDIVIDUAL_SHOW_SCORE_DENSITY_HISTOGRAM==1
    MAXIMUM=0;
    for m=1:Number_of_images

        IMAGE_NAME=listing_image(m).name;

NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
IMAGE_',IMAGE_NAME,'.mat'];
        A=load(NAME,'A');
        A=A.A;

        if max(max(A))>MAXIMUM
            MAXIMUM= max(max(A));
        end

    end

    RELATIVE=MAXIMUM;

  for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
IMAGE_',IMAGE_NAME,'.mat'];
        A=load(NAME,'A');
        A=A.A;



        SPLIT=256/NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM;

        [h,j] = find((round(RELATIVE*HIGH_BOUND))<=A);
        for m = 1:length(h),
            H(h(m),j(m)) = 256;
        end;

        for k=1:1:(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-3);
            k;

            [h,j]     =     find((round(RELATIVE*HIGH_BOUND/(Decimation_factor^k)))<=A     &
A<(round(RELATIVE*HIGH_BOUND/(Decimation_factor^(k-1)))));
            for m = 1:length(h),
                H(h(m),j(m))                                                              =
round(SPLIT*(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-k));  %  Those  bins  in  A
having 76-150 hits are reassigned to a pixel intensity value of 226
            end;
```

```
        end

        [h,j] = find(A==2);
        for m = 1:length(h),
            H(h(m),j(m)) = round(SPLIT*2);
        end;

        [h,j] = find(A==1);
        for m = 1:length(h),
            H(h(m),j(m)) = round(SPLIT*1);
        end;



        [h,j] = find(A==0);
        for m = 1:length(h),
            H(h(m),j(m)) = 0;
        end;


        colormap(hot(256));
        map = colormap;

        handle=figure(1);
        imshow(H,map,'InitialMagnification','fit');%
        xlabel(num2str(COMPOSANTE_NUM_1));
        ylabel(num2str(COMPOSANTE_NUM_2));


filename_figure=['SCORE_DENSITY_FIGURE_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANT
E_NUM_2),'_IMAGE_',IMAGE_NAME];

        print(handle,'-djpeg',filename_figure);


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_IMAGE_',IMAGE_NAME,'.mat'];
        save(NAME,'H');

    end

end
%%

if MASK_DATA_GENERATION==1


    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);



    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);

    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;
        MATRIX_NAME=strrep(IMAGE_NAME,IMAGE_FORMAT, '');
```

```
        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);

        SPECTRAL=double(IMAGE);

        [a,b,c]=size(SPECTRAL);
        t1new=[];
        t2new=[];

        for i=1:b

            X =reshape(SPECTRAL(:,i,:),a,c)';
            t1=X'*PT(:,COMPOSANTE_NUM_1);
            t2=X'*PT(:,COMPOSANTE_NUM_2);


            s1  =  ones(size(t1,1),1)+round(((t1  -  (t1min*ones(size(t1,1),1)))./((t1max-
t1min))).*(nbrgroupes-1));
            s2  =  ones(size(t2,1),1)+round(((t2  -  (t2min*ones(size(t2,1),1)))./((t2max-
t2min))).*(nbrgroupes-1));

            t1new=[t1new s1'];
            t2new=[t2new s2'];




        end



        NAME=['T',num2str(COMPOSANTE_NUM_1),'_NEW_','_IMAGE_',MATRIX_NAME,'.mat'];
        save(NAME,'t1new');
        NAME=['T',num2str(COMPOSANTE_NUM_2),'_NEW_','_IMAGE_',MATRIX_NAME,'.mat'];
        save(NAME,'t2new');
    end

end


%%
if MASK_FINDER_ROIPOLY==1


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'.mat'];
    H=load(NAME,'H');
    H=H.H;

    colormap(hot(256));
    map = colormap;
    HRGB = uint8(round(ind2rgb(H,map)*255));


    criteria=1;

    BW=zeros(nbrgroupes,nbrgroupes);

    while criteria==1


        figure(1),imshow(HRGB);
        BW_inter=roipoly;

        BW=BW+BW_inter;

        criteria=input('Do you wish to add another mask? Yes-1 No-0  ?');

    end
```

```
NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_MASK_',num2str(MASK_NUMBER),'.mat'];
    save(NAME,'BW');

    if INVERSE_MASKING==1

NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_MASK_',num2str(MASK_NUMBER),'.mat'];
        BW=~BW;
        save(NAME,'BW');
        BW=~BW;
    end

    [aaa,bbb]=size(BW);
    for ii=1:aaa
        for jj=1:bbb
            if INVERSE_MASKING==0
                if BW(ii,jj)==1
                    HRGB(ii,jj,1)=0;
                    HRGB(ii,jj,2)=191;
                    HRGB(ii,jj,3)=255;
                end
            end

            if INVERSE_MASKING==1

                if BW(ii,jj)==0
                    HRGB(ii,jj,1)=0;
                    HRGB(ii,jj,2)=191;
                    HRGB(ii,jj,3)=255;
                end
            end

        end
    end

    handle2=figure(2);
    imshow(HRGB);


filename_figure=['SCORE_DENSITY_MASK_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_
NUM_2),'_MASK_',num2str(MASK_NUMBER),'.jpg'];

    print(handle2,'-djpeg',filename_figure);



    [MASK_ROW,MASK_COLUMN]=find(BW==1);

    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;
        MATRIX_NAME=strrep(IMAGE_NAME,IMAGE_FORMAT, '');
        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);
        [a,b,c]=size(IMAGE);

        NAME=['T',num2str(COMPOSANTE_NUM_1),'_NEW_','_IMAGE_',MATRIX_NAME,'.mat'];
        t1new=load(NAME,'t1new');
        t1new=t1new.t1new;
        NAME=['T',num2str(COMPOSANTE_NUM_2),'_NEW_','_IMAGE_',MATRIX_NAME,'.mat'];
        t2new=load(NAME,'t2new');
        t2new=t2new.t2new;

        pixelmaskt1 = zeros(size(t2new));


        for a = 1:length(MASK_ROW)
            for b = 1:length(t1new)
                if MASK_ROW(a)==t2new(b) && MASK_COLUMN(a)==t1new(b)
```

```
                    pixelmaskt1(b) = 1;% On met des valeurs

                end
            end
        end
        [hauteurt1image,largeurt1image,lamda]=size(IMAGE);
        SEGMENTATION_BINARY = (reshape(pixelmaskt1,hauteurt1image,largeurt1image));

        IMAGE_OUT=double(IMAGE);
        [a,b,c]=size(IMAGE_OUT);
        for ii=1:a
            for jj=1:b
                if INVERSE_MASKING==0
                    if SEGMENTATION_BINARY(ii,jj)==1
                        IMAGE_OUT(ii,jj,1)=0;
                        IMAGE_OUT(ii,jj,2)=191;
                        IMAGE_OUT(ii,jj,3)=255;
                    end
                end

                if INVERSE_MASKING==1
                    if SEGMENTATION_BINARY(ii,jj)==0
                        IMAGE_OUT(ii,jj,1)=0;
                        IMAGE_OUT(ii,jj,2)=191;
                        IMAGE_OUT(ii,jj,3)=255;
                    end
                end
            end
        end

        handle=figure(3);
        imshow(uint8(IMAGE_OUT));


filename_figure=['SCORE_DENSITY_FIGURE_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANT
E_NUM_2),'_IMAGE_',IMAGE_NAME,'_MASK_',num2str(MASK_NUMBER),'.jpg'];

        print(handle,'-djpeg',filename_figure);


    end

end


%%
if X_MATRIX_GENERATION==1

    X=[];

    for m=1:Number_of_images
        m

        IMAGE_NAME=listing_image(m).name;
        MATRIX_NAME=strrep(IMAGE_NAME,IMAGE_FORMAT, '');
        IMAGE=imread([directory_name_image,'\',IMAGE_NAME]);
        [a,b]=size(IMAGE);


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
IMAGE_',MATRIX_NAME,'.mat'];
        A=load(NAME,'A');
        A=A.A;

        X=[X;reshape(A,1,nbrgroupes*nbrgroupes)];

    end
    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    save(NAME,'X');
```

```
end

%%
if Y_MATRIX_GENERATION==1

    [num,txt,raw]= xlsread(Y_MATRIX_SOURCE);

    Y=num;
    Y_VARIABLES=txt;
    %Y_VARIABLES=[cellstr(['Variable 1']),cellstr(['Variable 2'])];


    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    save(NAME,'Y');
    save('Y_VARIABLES.mat','Y_VARIABLES');
end

%%
% Covariance matrix between X and y

if COVARIANCE_X_Y==1

    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    X=load(NAME,'X');
    X=X.X;

    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    Y=load(NAME,'Y');
    Y=Y.Y;
    load('Y_VARIABLES.mat');
    [a,b]=size(X);

    [c,d]=size(Y);


    for i=1:d

        COVARIANCE=[];

        for j=1:b

            C=cov(X(:,j),Y(:,i));

            COVARIANCE=[COVARIANCE,C(2,1)];

        end

        COVARIANCE=reshape(COVARIANCE,nbrgroupes,nbrgroupes);

        colormap(jet);
        map = colormap;

        %
        COVARIANCE_NEGATIVE=COVARIANCE;
        COVARIANCE_POSITIVE=COVARIANCE;

        COVARIANCE_NEGATIVE(COVARIANCE>0)=0;
        COVARIANCE_NEGATIVE=abs(COVARIANCE_NEGATIVE);
        COVARIANCE_POSITIVE(COVARIANCE<0)=0;


COVARIANCE_NEGATIVE=round(COVARIANCE_NEGATIVE/max(max(COVARIANCE_NEGATIVE)).*255);

COVARIANCE_POSITIVE=round(COVARIANCE_POSITIVE/max(max(COVARIANCE_POSITIVE)).*255);

        COVARIANCE_NEGATIVE=label2rgb(round(COVARIANCE_NEGATIVE),'winter','k');
        COVARIANCE_POSITIVE=label2rgb(round(COVARIANCE_POSITIVE),'autumn','k');

        COVARIANCE_OUT=COVARIANCE_NEGATIVE+COVARIANCE_POSITIVE;
        %
```

```
        filename_figure=['COVARIANCE_MATRIX_',char(Y_VARIABLES(i))];
        handle=figure(2);
        %imshow(COVARIANCE_OUT,map,'InitialMagnification','fit');%
        imshow(COVARIANCE_OUT,'InitialMagnification','fit');%

        print(handle,'-djpeg',filename_figure);


    end

end

%%

if PHASE_ANGLE==1

    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    X=load(NAME,'X');
    X=X.X;

    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    Y=load(NAME,'Y');
    Y=Y.Y;
    load('Y_VARIABLES.mat');
    [a,b]=size(X);
    [c,d]=size(Y);


    %Computing Z1 and Z2 (see Honglu Yu, John F. MacGregor 2003)

    Z1=Y(:,1);

    minY=min(Y(:,1));
    maxY=max(Y(:,1));

    ystar=(minY+maxY)/2;

    Z2=abs(Z1-ystar);

    COVARIANCE_1=[];
    for j=1:b

        C=cov(X(:,j),Z1);

        COVARIANCE_1=[COVARIANCE_1,C(2,1)];

    end
    COVARIANCE_1=reshape(COVARIANCE_1,nbrgroupes,nbrgroupes);


    COVARIANCE_2=[];
    for j=1:b

        C=cov(X(:,j),Z2);

        COVARIANCE_2=[COVARIANCE_2,C(2,1)];

    end
    COVARIANCE_2=reshape(COVARIANCE_2,nbrgroupes,nbrgroupes);


    Theta=atan2(COVARIANCE_1,COVARIANCE_2);

    COVARIANCE_NEGATIVE=Theta;
    COVARIANCE_POSITIVE=Theta;

    COVARIANCE_NEGATIVE(Theta>0)=0;
    COVARIANCE_NEGATIVE=abs(COVARIANCE_NEGATIVE);
    COVARIANCE_POSITIVE(Theta<0)=0;

    COVARIANCE_NEGATIVE=round(COVARIANCE_NEGATIVE/max(max(COVARIANCE_NEGATIVE)).*255);
```

```
    COVARIANCE_POSITIVE=round(COVARIANCE_POSITIVE/max(max(COVARIANCE_POSITIVE)).*255);

    COVARIANCE_NEGATIVE=label2rgb((COVARIANCE_NEGATIVE),'winter','k');
    COVARIANCE_POSITIVE=label2rgb((COVARIANCE_POSITIVE),'autumn','k');

    COVARIANCE_OUT=uint8(COVARIANCE_NEGATIVE+COVARIANCE_POSITIVE);
    %

    filename_figure=['THETA_MATRIX'];
    handle=figure(3);
    %imshow(COVARIANCE_OUT,map,'InitialMagnification','fit');%
    imshow(COVARIANCE_OUT,'InitialMagnification','fit');%

    print(handle,'-djpeg',filename_figure);

    MODULE=sqrt(COVARIANCE_1.^2+COVARIANCE_2.^2);

    COVARIANCE_POSITIVE=MODULE;

    COVARIANCE_POSITIVE=round(COVARIANCE_POSITIVE/max(max(COVARIANCE_POSITIVE)).*255);

    COVARIANCE_POSITIVE=label2rgb((COVARIANCE_POSITIVE),'autumn','k');

    COVARIANCE_OUT=uint8(COVARIANCE_POSITIVE);
    %

    filename_figure=['MODULE_MATRIX'];
    handle=figure(4);
    %imshow(COVARIANCE_OUT,map,'InitialMagnification','fit');%
    imshow(COVARIANCE_OUT,'InitialMagnification','fit');%

    print(handle,'-djpeg',filename_figure);



end

%%

if HISTOGRAMS_GENERATION==1

    HISTOGRAMS_COMPILED=[];

    for m=1:Number_of_images

        IMAGE_NAME=listing_image(m).name;

NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
IMAGE_',IMAGE_NAME,'.mat'];
        A=load(NAME,'A');
        A=A.A;

        [aa,bb]=size(A);

        A_T2=sum(A,2);

        A_T2_prob=A_T2./sum(sum(A));

         HISTOGRAMS_COMPILED=[HISTOGRAMS_COMPILED;A_T2_prob'];

    end

    HISTOGRAMS_COMPILED_SUMMED=zeros(Number_of_images,bb);
    HISTOGRAMS_COMPILED_SUMMED(:,1)=HISTOGRAMS_COMPILED(:,1);

    for i=2:bb
         HISTOGRAMS_COMPILED_SUMMED(:,i)=HISTOGRAMS_COMPILED_SUMMED(:,i-
1)+HISTOGRAMS_COMPILED(:,i);

    end
```

```
    filename_figure=['PROBABILITY_HISTOGRAM'];
    handle=figure(1);
    plot(HISTOGRAMS_COMPILED'),legend(listing_image.name);
    print(handle,'-djpeg',filename_figure);

    filename_figure=['SUMMED_HISTOGRAM'];
    handle=figure(2);
    plot(HISTOGRAMS_COMPILED_SUMMED'),legend(listing_image.name);
    print(handle,'-djpeg',filename_figure);



end



%%
if PROMV_EXPORT==1


    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    X=load(NAME,'X');
    X=X.X;

    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    Y=load(NAME,'Y');
    Y=Y.Y;
    load('Y_VARIABLES.mat');
    [a,b]=size(X);

    [c,d]=size(Y);

    Nb=2;

    obsIDs=cell(a,1);

    num=1;
    for i=1:a
        obsIDs(i,1)={num2str(num)};
        num=num+1;
    end

    BlockNames=cell(Nb,1);

    BlockNames(1,1)={'X'};
    BlockNames(2,1)={'Y'};

    BlockTypes=cell(Nb,1);

    BlockTypes(1,1)={'normal'};
    BlockTypes(2,1)={'normal'};


    X_VARIABLES_STRINGS=cell(b,1);


    for j=1:b
        X_VARIABLES_STRINGS(j,1)={['SCORE_DENSITY_HISTOGRAM_BIN_',num2str(j)]};

    end


    varIDs=cell(Nb,1);

    varIDs(1,1)={X_VARIABLES_STRINGS};
    varIDs(2,1)={Y_VARIABLES'};


    DataBlocks=cell(Nb,1);

    DataBlocks(1,1)={X};
```

```
    DataBlocks(2,1)={Y};

    ObservationLengths=cell(Nb,1);
    ObservationLengths(1,1)={'NaN'};
    ObservationLengths(2,1)={'NaN'};


    NAME=['PROMV_EXPORT',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];

save(NAME,'BlockNames','BlockTypes','varIDs','DataBlocks','ObservationLengths','obsIDs');


end




%%
if CROSSVAL==1

    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    X=load(NAME,'X');
    X=X.X;

    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    Y=load(NAME,'Y');
    Y=Y.Y;
    load('Y_VARIABLES.mat');


    pre = {preprocess('autoscale') preprocess('autoscale')};
    opts.preprocessing = pre;

    [press,cumpress] = crossval(X,Y,'sim',{'loo'},10,opts);

end

%%
if PLS_MODEL==1

    NAME=['X_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    X=load(NAME,'X');
    X=X.X;

    NAME=['Y_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.mat'];
    Y=load(NAME,'Y');
    Y=Y.Y;
    load('Y_VARIABLES.mat');


    [rx,cx]=size(X);
    [rYresp cYresp]=size(Y);
    pre={['autoscale'] ['autoscale']};
    opts.preprocessing = pre;
    opts.weights=['hist'];
    opts.weights=[];
    opts.display='off';
    opts.plots= 'none';
    opts.outputversion=1;
    opts.algorithm=['sim'];

    model=pls(X,Y,NUMBER_OF_COMPONENTS_PLS,opts);

    Y_pred=model.pred{1,2};

    Y_mean=mean(Y,1);

    [a,b]=size(Y);
    SST=0;
    SSE=0;
    for i=1:a
```

```
        SST=SST+(Y(i,:)-Y_mean).^2;
        SSE=SSE+(Y(i,:)-Y_pred(i,:)).^2;
    end
    R2=(ones(1,b)-SSE./SST);


    for j=1:b

    figure(j),scatter(Y(:,j),Y_pred(:,j));

    end

end
```

Appendix G. Matlab® scripts used for hyperspectral microscopy

POINTS_GENERATOR_METAMORPH_v0r1.m

```
%This m-file creates a text file to load points in Metamorph. The points
%form a grid. There is one grid (one text file) per well.

%The .STG files must be put in C:\MM\app\mmproc\DATA

%Those files can be used with the MDA application in Metamorph (for
%scanning multiwell plates).

%IMPORTANT NOTE: The position (0,0) of the stage must be set at the center
%of the first well (top left corner).


%%Author: Pierre-Marc Juneau
%%Date: August 2010

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%INPUTS%%

%Basename for the .STG files

BASENAME='PMJ_HYPERSPECTRAL_20X_';

%Intervals (in um) between images (depending on focus):
%10X: Delta_X=1.0, Delta_Y=0.8
%20X: Delta_X=0.5, Delta_Y=0.4
%40X: Delta_X=0.25, Delta_Y=0.2

Delta_X=0.35/1000;
Delta_Y=1;
Corners=-2; %The Corners option could be used to "chop" some unwanted points at the
corners

%Distances between wells (depending on type of multiwell plate, in mm):
%6 well plate: 40
%12 well plate: 26
%24 well plate: 19.4
%96 well plate: 9.0

Bias_X=40;
Bias_Y=Bias_X;

%Number of rows and columns in the multiwell plate:
```

```
n_Rows=2;
n_Colums=3;

%Grid dimensions (depending on acquisition time wanted):
%For example, for an 11x11 grid, we have X_HEIGHT=11 and Y_HEIGHT=11
%Note that those must be odd numbers!

X_HEIGHT=1001;
Y_HEIGHT=1;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initialisation of the focus matrix
Number_points=X_HEIGHT*Y_HEIGHT;
z=zeros(Number_points,1);

nX=floor(X_HEIGHT/2);
nY=floor(Y_HEIGHT/2);

HEIGHT_X=nX*Delta_X;
HEIGHT_Y=nY*Delta_Y;

Well_number=1;

%For loop for printing the files

for ii=1:n_Rows
    for jj=1:n_Colums


BiasX=(jj-1)*Bias_X;
BiasY=(ii-1)*Bias_Y;

pos_y_high=HEIGHT_Y+BiasY;
pos_y_low=-HEIGHT_Y+BiasY;
pos_x_high=HEIGHT_X+BiasX;
pos_x_low=-HEIGHT_X+BiasX;


filename=[BASENAME num2str(Well_number) '.STG']

fid=fopen(filename,'wt');

fprintf(fid,'"Stage Memory List", Version 5.0\n');
fprintf(fid,'0, 0, 0, 0, 0, 0, 0, "mm", "mm"\n');
fprintf(fid,'0\n');
fprintf(fid,'%0.0f\n',Number_points);
```

```
a=1;

for i=pos_y_high:-Delta_Y:pos_y_low
    for j=pos_x_low:Delta_X:pos_x_high

        a
        perm1=0;
        perm2=0;

        if i<(pos_y_high-Corners) && i>(pos_y_low+Corners)
            perm1=1;
        end
        if j<(pos_x_high-Corners) && j>(pos_x_low+Corners)
            perm2=1;
        end
        if perm1==1 || perm2==1;
        fprintf(fid,'"%0.0f", %0.5f, %0.5f, %0.5f, 0, 0, FALSE, -9999, TRUE, TRUE,
0\n',a,j,i,z(a));
         a=a+1;
        end
    end
end
fclose(fid);
Well_number=Well_number+1;

    end
end
```

T1_generation.m

```
%Script to compute a T1 score image on hypespectral microscopy hypercube
%Script modified by Pierre-Marc Juneau (2010)
%Adapted from LineScan2Image (Ryan Gosselin, 2006)
%

% START SCRIPT -------------------------
clc
clear all

% Start file select --------------------


% Inable script to work with bmp or png images differently.
dname = uigetdir('Select Directory for images to analyze');
Folder = dname;
Files = struct2cell(dir([Folder,'/', '*', '.TIF']))';
[l,c] = size(Files);


NumberOfImages = 0;

for i = 1:l
    if Files{i,4} == 0 % Ignore subfolders
        NumberOfImages = NumberOfImages + 1;
        FilteredFiles(NumberOfImages,:) = Files(i,1:3);
        %FilteredFiles{NumberOfImages,2} = datestr(datenum(Files{i,2}),31);
        %FilteredFiles{NumberOfImages,3} = datenum(Files{i,2});
    end
end

% 1=sort by name, 2=sort by date/time
SortedFiles = sortrows(Files, 2);

% Determining camera used = N:NIR, U:UV, V:VIS
Camera = SortedFiles{1,1};


disp(['Folder contains ' num2str(NumberOfImages) ' image(s).'])


Z = zeros(1024);

for i=1:NumberOfImages
    i
    ImageFileName = [Folder,'/', SortedFiles{i,1}];
```

```
    RGB = imread(ImageFileName);
    X = (double(RGB))';
    Z = Z + (X'*X);
end

[P,E,PT]=svd(Z);
variancecaptured = sum(E)./sum(sum(E));
variancecaptured1_10 = variancecaptured(1,1:10)

save('matriceZ.mat','Z')

%We will determine the t1 score image

p1=PT(:,1);

GLOBAL=[];

for i=1:NumberOfImages
    aa=i;
    aa
    ImageFileName = [Folder,'/', SortedFiles{i,1}];
    RGB = imread(ImageFileName);
    X = (double(RGB))';
    T1part=X*p1;
    GLOBAL =[GLOBAL T1part];
end

RRR=GLOBAL./min(min(GLOBAL))*256;

TTT1=flipud(uint8(RRR));

save('matriceT1.mat','TTT1')

figure(1);
imshow(TTT1)
```

HYPERSPECTRAL_MIA_v0r3.m

```
%This function is used to perform MIA on NIR hyperspectral microscopy images

%Pierre-Marc Juneau, February 2014

clear
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%INPUTS

Folder='E:\Experiment_January_06_2013\NIR\Dummy frames\';
Folder_NIR='E:\Experiment_January_06_2013\NIR\NIR_SHOTS\2013-01-0614_19_24';
BASE_NAME_1='BB13MP5_NIR_';
BASE_NAME_2='_s';
NUMBER_OF_WELLS=6;
NUMBER_OF_LINESCANS=[1001 1001 1001 1001 1001 1001];
LAMBDA=256;
NIR=1;
NIR_CORRECTION=1;

MAPPING_NIR=0;
COMPUTE_AVERAGE_NIR_BACKGROUND=0;
LATENT_SPACE_GENERATION=0;
MINIMUM_MAXIMUM_SCORES_CALCULATION=0;
SCORE_IMAGE_COMPUTATION=0;
SCORE_DENSITY_HISTOGRAM_MATRIX=0;
INDIVIDUAL_SCORE_DENSITY_HISTOGRAM_MATRIX=0;
SHOW_SCORE_DENSITY_HISTOGRAM=0;
INDIVIDUAL_SHOW_SCORE_DENSITY_HISTOGRAM=0;
MASK_FINDER_ROIPOLY=1;



COMPOSANTE_NUM_1=2;
COMPOSANTE_NUM_2=3;

MASK_NUMBER=8;
INVERSE_MASKING=0;


nbrgroupes=256;
%Paramenters for score density histogram display
NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM=10;
Decimation_factor=10;
HIGH_BOUND=0.5;

alpha=0.5; %Alpha value for F-Berkeley function
MIN_SCORE=135;
MAX_SCORE=150;
window_delta=5;

IMAGE_FORMAT='.TIF';

listing_image = dir([Folder_NIR,'\',['*','.PNG']]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Generating common latent space for all images

if MAPPING_NIR==1

    MAP_NIR=zeros(NUMBER_OF_LINESCANS(1),NUMBER_OF_WELLS);

    Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    [l,c]=size(Files);
```

```
    NIR_DATES=[];

    for i = 1:l
        if Files{i,4} == 0 % Ignore subfolders
            NIR_DATES=[NIR_DATES;datenum(Files{i,2})];
        end
    end



    for m=1:NUMBER_OF_WELLS
        m

        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];
            Carac=struct2cell(dir(ImageFileName));
            Date_time=datenum(Carac{2});

            tmp = abs(NIR_DATES-Date_time);
            [idx idx] = min(tmp);

            MAP_NIR(k,m)=idx;


        end

    end
    save('MAP_NIR.mat','MAP_NIR');

end

if COMPUTE_AVERAGE_NIR_BACKGROUND==1

    dname = uigetdir('Select Directory for images to analyze');
    Folder = dname;
    Files = struct2cell(dir([Folder,'/', '*', '.PNG']))';
    [l,c] = size(Files);

    NumberOfImages = 0;

    for i = 1:l
        if Files{i,4} == 0 % Ignore subfolders
            NumberOfImages = NumberOfImages + 1;
            FilteredFiles(NumberOfImages,:) = Files(i,1:3);
            %FilteredFiles{NumberOfImages,2} = datestr(datenum(Files{i,2}),31);
            %FilteredFiles{NumberOfImages,3} = datenum(Files{i,2});
        end
    end

    % 1=sort by name, 2=sort by date/time
    SortedFiles = sortrows(Files, 2);


    for i=1:NumberOfImages
        i
        if i==1

            ImageFileName = [Folder,'/', SortedFiles{i,1}];
            RGB = imread(ImageFileName);
            SUM=double(RGB);

        else


            ImageFileName = [Folder,'/', SortedFiles{i,1}];
            RGB = imread(ImageFileName);
            SUM = SUM + double(RGB);
```

```
        end

    end

    AVERAGE=SUM./NumberOfImages;

    %Need to remove zeros to avoid problems

    MEAN_AVERAGE=mean(mean(AVERAGE));

    AVERAGE(AVERAGE==0)=MEAN_AVERAGE;

    save('AVERAGE.mat','AVERAGE');


end

%%

if LATENT_SPACE_GENERATION==1


    Z=zeros(LAMBDA,LAMBDA);


    if NIR==1

        MAP_NIR=load('MAP_NIR.mat','MAP_NIR');
        MAP_NIR=MAP_NIR.MAP_NIR;

        AVERAGE=load('AVERAGE.mat','AVERAGE');
        AVERAGE=AVERAGE.AVERAGE;

        Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    end

    for m=1:NUMBER_OF_WELLS
        m

        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];

            if NIR==1

                IDENTIFIER=MAP_NIR(k,m);

                ImageFileName = [Folder_NIR,'\',Files{IDENTIFIER,1};];

            end

            RGB = imread(ImageFileName);

            if NIR_CORRECTION==1

                RGB=double(RGB)./AVERAGE;

            end


            X = (double(RGB))';
            Z = Z + (X'*X);

        end

    end
    save('Zmatrix.mat','Z');
```

```
end


%%
if MINIMUM_MAXIMUM_SCORES_CALCULATION==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);

    %%

    %We find the minimum and the maximum value of all the scores

    NUMBER_TOTAL_COMPONENTS=size(Z,1);

    SCORES_MIN=ones(1,LAMBDA).*realmax;
    SCORES_MAX=ones(1,LAMBDA).*(-realmax);

    if NIR==1

        MAP_NIR=load('MAP_NIR.mat','MAP_NIR');
        MAP_NIR=MAP_NIR.MAP_NIR;

        AVERAGE=load('AVERAGE.mat','AVERAGE');
        AVERAGE=AVERAGE.AVERAGE;

        Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    end

    for m=1:NUMBER_OF_WELLS
        m

        GLOBAL_1=[];
        GLOBAL_2=[];


        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];

            if NIR==1

                IDENTIFIER=MAP_NIR(k,m);

                ImageFileName = [Folder_NIR,'\',Files{IDENTIFIER,1};];

            end

            RGB = imread(ImageFileName);

             if NIR_CORRECTION==1

                 RGB=double(RGB)./AVERAGE;

             end

            X = (double(RGB))';

            T=X*PT;

            SCORES_LOCAL_MAX=max(T);
            SCORES_LOCAL_MIN=min(T);

            for kk=1:LAMBDA
```

```
                if SCORES_LOCAL_MAX(kk)>SCORES_MAX(kk)
                    SCORES_MAX(kk)=SCORES_LOCAL_MAX(kk);
                end

                if SCORES_LOCAL_MIN(kk)<SCORES_MIN(kk)
                    SCORES_MIN(kk)=SCORES_LOCAL_MIN(kk);
                end

            end

        end


    end
    save('SCORES_MIN.mat','SCORES_MIN');
    save('SCORES_MAX.mat','SCORES_MAX');

end


%%
if SCORE_IMAGE_COMPUTATION==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);

    %%

    %We find the minimum and the maximum value of all the scores

    if NIR==1

        MAP_NIR=load('MAP_NIR.mat','MAP_NIR');
        MAP_NIR=MAP_NIR.MAP_NIR;

        AVERAGE=load('AVERAGE.mat','AVERAGE');
        AVERAGE=AVERAGE.AVERAGE;

        Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    end

    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);

   for m=1:NUMBER_OF_WELLS
       m

       GLOBAL_1=[];
       GLOBAL_2=[];


        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];

            if NIR==1
```

```
                IDENTIFIER=MAP_NIR(k,m);

                ImageFileName = [Folder_NIR,'\',Files{IDENTIFIER,1};];

            end


            RGB = imread(ImageFileName);

             if NIR_CORRECTION==1

                 RGB=double(RGB)./AVERAGE;

             end


            X = (double(RGB))';


            t1=X*PT(:,COMPOSANTE_NUM_1);
            t2=X*PT(:,COMPOSANTE_NUM_2);

            GLOBAL_1 =[GLOBAL_1 flipud((t1))];
            GLOBAL_2 =[GLOBAL_2 flipud((t2))];


        end

        GLOBAL_1=uint16(((GLOBAL_1-t1min)./(t1max-t1min)).*65535);
        GLOBAL_2=uint16(((GLOBAL_2-t2min)./(t2max-t2min)).*65535);

        %GLOBAL_1=label2rgb(GLOBAL_1,'Grey');
        %GLOBAL_2=label2rgb(GLOBAL_2,'Grey');

        handle=figure(1);
        imshow(histeq(GLOBAL_1),[]);

filename_figure=['SCORE_IMAGE_','T',num2str(COMPOSANTE_NUM_1),'_Well_',num2str(m)];
        print(handle,'-djpeg',filename_figure);

NAME=['SCORE_IMAGE_MATRIX_','T',num2str(COMPOSANTE_NUM_1),'_Well_',num2str(m),'.mat'];
        GLOBAL=GLOBAL_1;
        save(NAME,'GLOBAL');

        handle=figure(2);
        imshow(histeq(GLOBAL_2),[]);

filename_figure=['SCORE_IMAGE_','T',num2str(COMPOSANTE_NUM_2),'_Well_',num2str(m)];
        print(handle,'-djpeg',filename_figure);

NAME=['SCORE_IMAGE_MATRIX_','T',num2str(COMPOSANTE_NUM_2),'_Well_',num2str(m),'.mat'];
        GLOBAL=GLOBAL_2;
        save(NAME,'GLOBAL');

   end

end


%%
%Computation of the density histogram matrix

if SCORE_DENSITY_HISTOGRAM_MATRIX==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);

    A = zeros(nbrgroupes,nbrgroupes);
```

```
    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);

    if NIR==1

        MAP_NIR=load('MAP_NIR.mat','MAP_NIR');
        MAP_NIR=MAP_NIR.MAP_NIR;

        AVERAGE=load('AVERAGE.mat','AVERAGE');
        AVERAGE=AVERAGE.AVERAGE;

        Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    end

    for m=1:NUMBER_OF_WELLS
        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];

            if NIR==1

                IDENTIFIER=MAP_NIR(k,m);

                ImageFileName = [Folder_NIR,'\',Files{IDENTIFIER,1};];

            end


            RGB = imread(ImageFileName);

             if NIR_CORRECTION==1

                 RGB=double(RGB)./AVERAGE;

             end


            X = (double(RGB))';


            t1=X*PT(:,COMPOSANTE_NUM_1);
            t2=X*PT(:,COMPOSANTE_NUM_2);



            s1  =  ones(size(t1,1),1)+round(((t1  -  (t1min*ones(size(t1,1),1)))./((t1max-
t1min))).*(nbrgroupes-1));
            s2  =  ones(size(t2,1),1)+round(((t2  -  (t2min*ones(size(t2,1),1)))./((t2max-
t2min))).*(nbrgroupes-1));

            for kk = 1:length(s1)

                A(s2(kk), s1(kk)) = A(s2(kk), s1(kk)) + 1;

            end


        end
    end
```

```
NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.
mat'];
    save(NAME,'A');
end


%%
if INDIVIDUAL_SCORE_DENSITY_HISTOGRAM_MATRIX==1

    Z=load('Zmatrix.mat','Z');
    Z=Z.Z;
    [P,E,PT]=svd(Z);
    %variancecaptured = sum(E)./sum(sum(E));
    %variancecaptured1_10 = variancecaptured(1,1:10);



    SCORES_MIN=load('SCORES_MIN.mat','SCORES_MIN');
    SCORES_MIN=SCORES_MIN.SCORES_MIN;

    SCORES_MAX=load('SCORES_MAX.mat','SCORES_MAX');
    SCORES_MAX=SCORES_MAX.SCORES_MAX;

    t1min=SCORES_MIN(COMPOSANTE_NUM_1);
    t2min=SCORES_MIN(COMPOSANTE_NUM_2);
    t1max=SCORES_MAX(COMPOSANTE_NUM_1);
    t2max=SCORES_MAX(COMPOSANTE_NUM_2);


    if NIR==1

        MAP_NIR=load('MAP_NIR.mat','MAP_NIR');
        MAP_NIR=MAP_NIR.MAP_NIR;

        AVERAGE=load('AVERAGE.mat','AVERAGE');
        AVERAGE=AVERAGE.AVERAGE;

        Files=struct2cell(dir([Folder_NIR,'/', '*', '.PNG']))';

    end

    for m=1:NUMBER_OF_WELLS
        A = zeros(nbrgroupes,nbrgroupes);
        for k=1:NUMBER_OF_LINESCANS(m)
            k

            ImageFileName                                                                    =
[Folder,BASE_NAME_1,num2str(m),BASE_NAME_2,num2str(k),IMAGE_FORMAT];

            if NIR==1

                IDENTIFIER=MAP_NIR(k,m);

                ImageFileName = [Folder_NIR,'\',Files{IDENTIFIER,1};];

            end


            RGB = imread(ImageFileName);

             if NIR_CORRECTION==1

                 RGB=double(RGB)./AVERAGE;

             end


            X = (double(RGB))';
```

```
            t1=X*PT(:,COMPOSANTE_NUM_1);
            t2=X*PT(:,COMPOSANTE_NUM_2);



            s1  =  ones(size(t1,1),1)+round(((t1  -  (t1min*ones(size(t1,1),1)))./((t1max-
t1min))).*(nbrgroupes-1));
            s2  =  ones(size(t2,1),1)+round(((t2  -  (t2min*ones(size(t2,1),1)))./((t2max-
t2min))).*(nbrgroupes-1));

            for kk = 1:length(s1)

                A(s2(kk), s1(kk)) = A(s2(kk), s1(kk)) + 1;

            end


        end

NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
Well_',num2str(m),'.mat'];
        save(NAME,'A');
    end
end



%%
%Showing the score density histogram

if SHOW_SCORE_DENSITY_HISTOGRAM==1


NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'.
mat'];
    A=load(NAME,'A');
    A=A.A;

    RELATIVE=max(max(A));

    SPLIT=256/NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM;

    [h,j] = find((round(RELATIVE*HIGH_BOUND))<=A);
    for m = 1:length(h),
        H(h(m),j(m)) = 256;
    end;

    for k=1:1:(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-3);
        k;

        [h,j]      =      find((round(RELATIVE*HIGH_BOUND/(Decimation_factor^k)))<=A      &
A<(round(RELATIVE*HIGH_BOUND/(Decimation_factor^(k-1)))));
        for m = 1:length(h),
            H(h(m),j(m))                                                                =
round(SPLIT*(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-k));  %  Those  bins  in  A
having 76-150 hits are reassigned to a pixel intensity value of 226
        end;


    end

    [h,j] = find(A==2);
    for m = 1:length(h),
        H(h(m),j(m)) = round(SPLIT*2);
    end;

    [h,j] = find(A==1);
    for m = 1:length(h),
        H(h(m),j(m)) = round(SPLIT*1);
    end;
```

```
    [h,j] = find(A==0);
    for m = 1:length(h),
        H(h(m),j(m)) = 0;
    end;


    colormap(hot(256));
    map = colormap;

    figure(1);
    imshow(H,map,'InitialMagnification','fit');%
    xlabel(num2str(COMPOSANTE_NUM_1));
    ylabel(num2str(COMPOSANTE_NUM_2));


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'.mat'];
    save(NAME,'H');


end

%%
if INDIVIDUAL_SHOW_SCORE_DENSITY_HISTOGRAM==1
    MAXIMUM=0;
    for m=1:NUMBER_OF_WELLS



NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
Well_',num2str(m),'.mat'];
        A=load(NAME,'A');
        A=A.A;

        if max(max(A))>MAXIMUM
            MAXIMUM= max(max(A));
        end

    end

    RELATIVE=MAXIMUM;

  for mm=1:NUMBER_OF_WELLS
        mm



NAME=['SCORE_DENSITY_MATRIX_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2),'_
Well_',num2str(mm),'.mat'];
        A=load(NAME,'A');
        A=A.A;



        SPLIT=256/NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM;

        [h,j] = find((round(RELATIVE*HIGH_BOUND))<=A);
        for m = 1:length(h),
            H(h(m),j(m)) = 256;
        end;

        for k=1:1:(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-3);
            k;

            [h,j]    =    find((round(RELATIVE*HIGH_BOUND/(Decimation_factor^k)))<=A    &
A<(round(RELATIVE*HIGH_BOUND/(Decimation_factor^(k-1)))));
            for m = 1:length(h),
```

```
                H(h(m),j(m))                                                                =
round(SPLIT*(NUMBER_OF_INTENSITY_LEVELS_SCORE_DENSITY_HISTOGRAM-k));  %  Those  bins  in  A
having 76-150 hits are reassigned to a pixel intensity value of 226
            end;


        end

        [h,j] = find(A==2);
        for m = 1:length(h),
            H(h(m),j(m)) = round(SPLIT*2);
        end;

        [h,j] = find(A==1);
        for m = 1:length(h),
            H(h(m),j(m)) = round(SPLIT*1);
        end;



        [h,j] = find(A==0);
        for m = 1:length(h),
            H(h(m),j(m)) = 0;
        end;


        colormap(hot(256));
        map = colormap;

        handle=figure(1);
        imshow(H,map,'InitialMagnification','fit');%
        xlabel(num2str(COMPOSANTE_NUM_1));
        ylabel(num2str(COMPOSANTE_NUM_2));


filename_figure=['SCORE_DENSITY_FIGURE_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANT
E_NUM_2),'_Well_',num2str(mm)];

        print(handle,'-djpeg',filename_figure);


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_Well_',num2str(mm),'.mat'];
        save(NAME,'H');

    end

end


%%
if MASK_FINDER_ROIPOLY==1


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'.mat'];
    H=load(NAME,'H');
    H=H.H;

    colormap(hot(256));
    map = colormap;
    HRGB = uint8(round(ind2rgb(H,map)*255));
    figure(1),imshow(HRGB);
    BW=roipoly;


NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_MASK_',num2str(MASK_NUMBER),'.mat'];
    save(NAME,'BW');

    if INVERSE_MASKING==1
```

```
NAME=['SCORE_DENSITY_HISTOGRAM_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_NUM_2)
,'_MASK_',num2str(MASK_NUMBER),'.mat'];
        BW=~BW;
        save(NAME,'BW');
        BW=~BW;
    end

    [aaa,bbb]=size(BW);
    for ii=1:aaa
        for jj=1:bbb
            if INVERSE_MASKING==0
                if BW(ii,jj)==1
                    HRGB(ii,jj,1)=0;
                    HRGB(ii,jj,2)=191;
                    HRGB(ii,jj,3)=255;
                end
            end

            if INVERSE_MASKING==1

                if BW(ii,jj)==0
                    HRGB(ii,jj,1)=0;
                    HRGB(ii,jj,2)=191;
                    HRGB(ii,jj,3)=255;
                end
            end

        end
    end

    handle2=figure(2);
    imshow(HRGB);


filename_figure=['SCORE_DENSITY_MASK_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANTE_
NUM_2),'_MASK_',num2str(MASK_NUMBER),'.jpg'];

    print(handle2,'-djpeg',filename_figure);



    [MASK_ROW,MASK_COLUMN]=find(BW==1);

    for m=1:NUMBER_OF_WELLS
        m



NAME=['SCORE_IMAGE_MATRIX_','T',num2str(COMPOSANTE_NUM_1),'_Well_',num2str(m),'.mat'];
        t1new=load(NAME,'GLOBAL');
        t1new=t1new.GLOBAL;
        t1new=floor(double(t1new)./65535.*255);

NAME=['SCORE_IMAGE_MATRIX_','T',num2str(COMPOSANTE_NUM_2),'_Well_',num2str(m),'.mat'];
        t2new=load(NAME,'GLOBAL');
        t2new=t2new.GLOBAL;
        t2new=floor(double(t2new)./65535.*255);

        [aaa,bbb]=size(t1new);

        pixelmaskt1 = zeros(aaa,bbb);


        for a = 1:length(MASK_ROW)
            for b = 1:aaa
                for c=1:bbb
                    if MASK_ROW(a)==t2new(b,c) && MASK_COLUMN(a)==t1new(b,c)

                        pixelmaskt1(b,c) = 1;% On met des valeurs
```

```
                end
            end
        end
    end


    IMAGE_OUT=label2rgb(uint8(t1new),'Gray');



    [a,b]=size(t1new);
    for ii=1:a
        for jj=1:b
            if INVERSE_MASKING==0
                if  pixelmaskt1(ii,jj)==1
                    IMAGE_OUT(ii,jj,1)=0;
                    IMAGE_OUT(ii,jj,2)=191;
                    IMAGE_OUT(ii,jj,3)=255;
                end
            end

            if INVERSE_MASKING==1
                if  pixelmaskt1(ii,jj)==0
                    IMAGE_OUT(ii,jj,1)=0;
                    IMAGE_OUT(ii,jj,2)=191;
                    IMAGE_OUT(ii,jj,3)=255;
                end
            end
        end
    end

    handle=figure(3);
    imshow(IMAGE_OUT);


filename_figure=['SCORE_DENSITY_FIGURE_T',num2str(COMPOSANTE_NUM_1),'T',num2str(COMPOSANT
E_NUM_2),'_Well_',num2str(m),'_MASK_',num2str(MASK_NUMBER),'.jpg'];

    print(handle,'-djpeg',filename_figure);


    end

end


%%
```