Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I

Generalized Least Squares-based Parametric Motion Estimation and Segmentation

Ph. D. THESIS

Presented by:
Raúl Montoliu Colás

Supervised by:
Dr. Filiberto Pla Bañón

Castellón, September 2008

Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I

Estimación y Segmentación de Modelos de Movimiento Paramétricos mediante Mínimos Cuadrados Generalizados

Tesis Doctoral

Presentada por:
Raúl Montoliu Colás

Dirigida por:
Dr. Filiberto Pla Bañón

Castellón, Septiembre de 2008

To Toya
To Ricardo, Rosita, Toni and Litín

Aportaciones, conclusiones y trabajo futuro

Resumen

El análisis del movimiento es uno de los campos más importantes de la visión por computador. Esto es debido a que el mundo real está en continuo movimiento y es obvio que podremos obtener mucha más información de escenas en movimiento que de escenas estáticas. El análisis del movimiento es una tarea fundamental para comprender el mundo en el que vivimos y es un requisito principal para crear cualquier tipo de mecanismo artificial que se desee que interactúe con su entorno. Uno de los objetivos del análisis del movimiento es crear un sistema artificial de percepción del movimiento que tenga un comportamiento similar al que poseemos los seres humanos. Aunque este proceso parece relativamente sencillo, al menos así nos lo parece a los humanos, es sobradamente conocido que a la hora de implementarlo en dispositivos artificiales presenta una dificultad enorme, principalmente debido a que todavía quedan aspectos de la visión humana que no han sido comprendidos en su totalidad.

El problema de estimar el movimiento de una determina región de una imagen es una de las tareas fundamentales del análisis del movimiento. En esta tesis se ha trabajado principalmente en desarrollar algoritmos de estimación de movimiento para su aplicación a problemas de registrado de imágenes y a problemas de segmentación del movimiento.

Es importante hacer una diferenciación respecto a que nos referimos con estimación de movimiento global y con estimación de movimiento local. Por un lado, por global nos referimos a que el area de la que queremos estimar el movimiento es la imagen completa. Las técnicas de estimación de movimiento global se aplican principalmente a problemas de registrado de imágenes [Brown, 1992], [Zitova and Flusser, 2003]) el cual es uno de los principales conceptos de esta tesis. Por otro lado, cuando hablamos de estimación local, nos referimos a que el área donde queremos estimar el movimiento suele ser muy pequeña, llegando incluso a ser tan pequeña como un único pixel. En este último caso, a este tipo de estimación de movimiento se le conoce como técnicas para calcular el flujo óptico ([Barron et al., 1994], [Beauchemin and Barron, 1995]).

En esta tesis se ha trabajado desarrollando técnicas de estimación global de movimiento principalmente para resolver problemas de registrado de imágenes. La estimación (local) del flujo óptico no es uno de los temas de este trabajo. Uno de los principales objetivos de este trabajo es desarrollar una técnica de registrado de imágenes de gran exactitud y que sea capaz de realizar su labor incluso en la presencia de deformaciones de gran magnitud. El capitulo 2 está complemente dedicado a este tema. Es aquí donde se propone un nuevo algoritmo de estimación de movimiento global el cual es capaz de trabajar con deformaciones de gran magnitud tales como traslaciones, rotaciones, cambios de escala, cambios de iluminación globales, etc., manteniendo un elevado nivel de exactitud. Una de las claves para conseguir altos niveles de exactitud es la forma en la que se ha formulado el problema de estimación de movimiento. En la formulación propuesta, cada observación tiene asignado un peso que es calculado a partir de la información que proporcionan los gradientes de la imagen en dicha observación, el cual tendrá valores elevados si la observación es considerada como inlier y valores bajos si la observación es considerada como outlier.

Hay que tener en cuenta que, en una secuencia de dos imágenes, también existe la posibilidad de que existan cambios de iluminación no espacialmente uniformes, es decir que no afectan a todos los pixeles por igual. En el capitulo 3 se ha desarrollado una modificación de la técnica anterior añadiendo un modelo dinámico de formación de la imagen, gracias al cual es posible registrar imágenes donde se ha producido un cambio de iluminación no uniforme manteniendo altos niveles de exactitud y también manteniendo la capacidad de trabajar con grandes deformaciones.

Otro de los objetivos de esta tesis es trabajar en problemas de estimación y la segmentación del movimiento en secuencias de dos imágenes. Segmentar el movimiento consiste en agrupar todos los pixeles que tienen el mismo movimiento aparente en una escena. Es un proceso similar a la segmentación de imágenes estáticas, pero en este caso, en vez de usar criterios de color para agrupar los pixeles, se usan criterios de movimiento. El proceso de estimar y segmentar el movimiento de forma simultánea tiene el inconveniente de ser un problema tipo "¿Qué fue antes, el huevo o la gallina?". Por un lado, si ya tenemos dada la estimación del movimiento de todos los pixeles, es relativamente fácil agruparlos en regiones. Por otro lado, si lo que tenemos dada es la agrupación, es también relativamente sencillo estimar el movimiento de cada grupo. El problema reside en obtener ambas cosas, la estimación y la segmentación, de forma simultánea.

El capitulo 4 de esta tesis está dedicado a este problema donde se presenta nuestro algoritmo el cual es capaz de segmentar y estimar el movimiento en una secuencia de dos imágenes de forma casi simultánea y sin conocimiento a priori del número de modelos de movimiento presentes. Para estimar el movimiento se usa el estimador desarrollado en el capitulo 2.

Objetivos

Como se ha comentado anteriormente, en esta tesis se ha trabajado principalmente desarrollando técnicas de estimación del movimiento global para su aplicación a problemas de registrado de imágenes y en desarrollar técnicas de estimación y segmentación simultanea del movimiento.

Los objetivos de este trabajo son los siguientes:

- Estudiar los principales algoritmos de estimación de movimiento prestando especial atención a aquellos que son usados en problemas de registrado de imágenes y para segmentar el movimiento.
- Diseñar algoritmos de estimación de movimiento que obtengan estimaciones de gran exactitud, aunque en la escena se encuentren outliers. Aplicarlos a problemas de registrado de imágenes y segmentación del movimiento
- Estudiar el problema de la presencia de deformaciones de gran magnitud y aportar soluciones para obtener estimaciones de gran exactitud, a pesar de dichas deformaciones.
- Estudiar el problema de la estimación y segmentación del movimiento simultánea.
- Diseñar un método de estimación y segmentación del movimiento que sea capaz de realizar la tarea de forma simultánea y sin conocer a priori el número de modelos que se encuentran en la escena.

Aportaciones y conclusiones

A continuación se comentan las principales contribuciones y conclusiones que se ha obtenido de este trabajo:

- **Estimación del movimiento**: Con respecto al problema de la estimación de movimiento, en este trabajo se ha explicado el problema y la diferencia entre estimación global y local. Puesto que es la estimación global la que más interés tiene en este trabajo, se ha revisado las principales técnicas que se pueden encontrar en la literatura. Dos de ellas han sido seleccionadas para ser comparadas contra nuestra propuesta.

- **Estimación global del movimiento mediante mínimos cuadrados generalizados**: Se ha estudiado la técnica de estimación GLS (Generalized Least Squares) para aplicarla a problemas de estimación de movimiento global. En este sentido, en este trabajo se ha propuesto una nueva técnica que ha sido aplicada con éxito a problemas de registrado de imágenes y de segmentación del movimiento. Una de las principales claves de nuestra propuesta es que la formulación que se ha realizado del problema proporciona una restricción adicional que ayuda al proceso de estimación ajustando los pixeles usando información del gradiente. Esto es conseguido gracias al uso de un peso para cada observación, el cual tendrá valores elevados en el caso de que la observación sea considerada como inlier y valores bajos cuando dicha observaciones sea considerado como outlier.

  Las principales características de nuestra propuesta son:

  - Nuestra propuesta usa un método de estimación no lineal basado en la técnica GLS. Por consiguiente, es posible usar directamente la BCA (Brightness Constancy Assumption) en vez de la ecuación del flujo óptico, proporcionando una aproximación al problema más cercana a la realidad.
  - Para evitar caer en un mínimo local, el algoritmo usa una técnica basada en características (en concreto usa la técnica SIFT, *Scale-Invariant Feature Transform* [Lowe, 2004]) mediante la cual se obtienen unos parámetros iniciales los cuales serán posteriormente refinados usando el estimador GLS para obtener mayor exactitud. Gracias a ello, el algoritmo propuesto es capaz de trabajar con grandes deformaciones a la vez que consigue estimaciones de gran exactitud.
  - El método de estimación GLS incluye en su diseño una restricción adicional mediante la cual es posible tratar con los outliers usando información del gradiente de la imagen. De forma similar a los métodos IRLS (Iteratively Reweight Least Squares), la restricción se expresa como un peso para cada observación.

La exactitud del método propuesto ha sido probada con imágenes reales de gran dificultad usando los modelos de movimiento afín y proyectivo. Para comparar nuestra propuesta se han seleccionado dos métodos que usan M-estimadores para tratar con los outliers y que están basados en una estrategia IRLS. Los resultados obtenidos demuestran que nuestra propuesta es capaz de obtener resultados tan buenos como los métodos basados en M-estimadores e incluso mejores en muchos casos.

- **Registrado de imágenes con deformaciones de gran magnitud**: Uno de los problemas principales a la hora de registrar dos imágenes ocurre cuando entre ambas existe una deformación de gran magnitud. En este trabajo se han revisado algunas de las más importantes aportaciones para tratar con este problema, la gran mayoría guardan relación con la extracción de características invariantes a rotaciones, cambios de escala, etc. En nuestra propuesta, hemos usado una técnica de extracción de características invariantes (basada en la técnica SIFT) la cual es aplicada en un primer paso del algoritmo propuesto, para obtener una primera estimación de los parámetros de movimiento reales, para en una segunda fase refinar la estimación mediante el estimador de movimiento propuesto basado en la técnica GLS.

- **Registrado de imágenes bajo cambios de iluminación no uniformes**: Otro de los problemas con los que nos podemos encontrar, son problemas de registrado en los que entre las imágenes existe un cambio de iluminación no espacialmente uniforme. Es decir, cambios de iluminación que no afectan a todos los pixeles, o que no afectan a todos los pixeles por igual. Para resolver estos casos, se ha estudiado el uso de un modelo dinámico de formación de imagen en el cual los factores de iluminación son funciones de la localización en vez de constantes, permitiendo obtener un modelo más general y exacto de cómo se forma la imagen. El uso de dicho modelo reemplaza a la BCA como función objetivo en nuestra propuesta de estimación de movimiento. Se han realizado una serie de experimentos que demuestran que el uso conjunto del estimador de movimiento propuesto junto con el modelo de imagen dinámico usado permite obtener estimaciones de gran exactitud a pesar de la presencia de fuertes cambios de iluminación.

- **Estimación y segmentación del movimiento**: Respecto al problema de estimación y segmentación del movimiento, los principales trabajos en esta materia han sido también revisados. Un nueva técnica para realizar esta tarea ha sido desarrollada. Nuestra propuesta usa como entrada secuencias de dos imágenes de niveles de gris y realiza su tarea sin conocer a priori el número de diferentes regiones en movimiento existen. El algoritmo usa información temporal mediante el estimador de movimiento propuesto e información espacial mediante un algoritmo iterativo de crecimiento de regiones el cual clasifica regiones de pixeles en sus correspondientes modelos de movimiento. Las principales características de nuestra propuesta son:

- El estimador de movimiento propuesto basado en GLS se usa para estimar el movimiento. Por lo tanto, se obtienen estimaciones de gran exactitud.
- El proceso de clasificación agrupa inliers, rechaza outliers e intercambia regiones entre los modelos, permitiendo mejorar la segmentación.
- El algoritmo propuesto usa regiones de pixeles en vez de pixeles aislados, así como información de vecindad, lo cual conlleva una mejor coherencia espacial.
- Después de que los modelos de movimiento han sido obtenidos, se aplica un proceso de refinado para afinar la segmentación a nivel de pixel.

Trabajo futuro

Aunque a lo largo de este trabajo se han realizado contribuciones interesantes, todavía queda mucho trabajo por hacer, ya sea para mejorar los algoritmos propuestos o para proponer otros nuevos. A continuación se presentan algunas líneas de trabajo futuro tanto a corto como a largo plazo:

- **Aumentar velocidad de proceso**: Los algoritmo presentados en este trabajo han sido cuidadosamente implementados. Sin embargo, es todavía posible mejorarlos en lo que se refiere a velocidad de proceso estudiando en profundidad si es posible evitar realizar algún cálculo secundario.

- **Probar los algoritmos usando otros modelos de movimiento**: En este trabajo se ha usado principalmente los modelos de movimiento afín y proyectivo. Existen problemas donde podría ser más conveniente usar otro tipos de modelos, como por ejemplo el modelo cuadrático. Los algoritmos desarrollados permiten de forma relativamente sencilla añadir nuevos modelos de movimiento.

- **Permitir deformación de mayor magnitud**: Aunque el grado de deformación con el algoritmo de registrado de imágenes es capaz de trabajar es muy alto, en un futuro se debería seguir estudiando nuevas técnicas que permitan trabajar con niveles de deformación todavía de mayor magnitud. En especial sería deseable aumentar el nivel de cambios de escala y permitir mayores cambios en el punto de vista.

- **Permitir mayor cambios de iluminación**: Como en el caso anterior, podría ser muy interesante estudiar técnicas que permitan registrar imágenes en presencia de cambios de iluminación de mayor magnitud.

- **Usar más de dos imágenes**: El método de segmentación del movimiento propuesto usa únicamente dos imágenes. Podría resultar interesante estudiar los efectos de usar más de dos imágenes lo que, probablemente, ayudaría en el proceso de segmentación.

Abstract

This thesis proposes several techniques related with the motion estimation problem. In particular, it deals with global motion estimation for image registration and motion segmentation. In the first case, we will suppose that the majority of the pixels of the image follow the same motion model, although the possibility of a large number of outliers are also considered. In the motion segmentation problem, the presence of more than one motion model will be considered. In both cases, sequences of two consecutive grey level images will be used.

A new generalized least squares-based motion estimator will be proposed. The proposed formulation of the motion estimation problem provides an additional constraint that helps to match the pixels using image gradient information. That is achieved thanks to the use of a weight for each observation, providing high weight values to the observations considered as inliers, and low values to the ones considered as outliers. To avoid falling in a local minimum, the proposed motion estimator uses a Feature-based method (SIFT-based) to obtain good initial motion parameters. Therefore, it can deal with large motions like translation, rotations, scales changes, viewpoint changes, etc.

The accuracy of our approach has been tested using challenging real images using both affine and projective motion models. Two Motion Estimator techniques, which use M-Estimators to deal with outliers into a iteratively reweighted least squared-based strategy, have been selected to compare the accuracy of our approach. The results obtained have showed that the proposed motion estimator can obtain as accurate results as M-Estimator-based techniques and even better in most cases.

The problem of estimating accurately the motion under non-uniform illumination changes will also be considered. A modification of the proposed global motion estimator will be proposed to deal with this kind of illumination changes.

In particular, a dynamic image model where the illumination factors are functions of the localization will be used replacing the brightens constancy assumption allowing for a more general and accurate image model. Experiments using challenging images will be performed showing that the combination of both techniques is feasible and provides accurate estimates of the motion parameters even in the presence of strong illumination changes between the images.

The last part of the thesis deals with the motion estimation and segmentation problem. The proposed algorithm uses temporal information, by using the proposed generalized least-squares motion estimation process and spatial information by using an iterative region growing algorithm which classifies regions of pixels into the different motion models present in the sequence. In addition, it can extract the different moving regions of the scene while estimating its motion quasi-simultaneously and without a priori information of the number of moving objects in the scene. The performance of the algorithm will be tested on synthetic and real images with multiple objects undergoing different types of motion.

Acknowledgements

First and foremost I want to express my gratitude and thanks to my thesis supervisor, Filiberto Pla for his guidance, accessibility, advice, patience, support, encouragement and friendship. I would also like to thank all my friends and colleges from the *Computer Vision Group* at this university: Javi, Pedro, Jorge, Salva, Jose Miguel, María ángeles, Jose, Ramón, Isabel, Gustavo, Tomás, Yasmina, Manoli, Adolfo, Nacho, Gemma and many other people that are not now working with us, for making our lab a pleasant and interesting place to work.

Thanks to the members of the *Computer Science and Engineering Department* and the *Computer Languages and System Department* at this university for helping in my teaching work. In particular, thanks to the members of the technical support of both departments for their work and their help in many technical problems.

Thanks to the members of the *Reconocimiento de Imágenes y Visión Artificial* at UPV's *Instituto Tecnológico de Informática* for their valuable help in my stay in their lab.

Thanks to the many authors who sent me papers, image databases, source code of their algorithms, or just for leaving their papers publicly available at their web pages.

Thanks to many people from latex, C++, openCv and Matlab newsgroups over Internet for solving all my doubts. The web sites `http:\www.webster.com` and `http:\www.wordreference.com` were also helpful in the process of writing this work in English.

Thanks to my family and friends, in particular to my wife and my parents and the rest of my family, for encouraging me and giving morale to continue with this work.

Thank to University Jaume I for providing me the framework and means for this thesis to be carried out.

Contents

Aportaciones, conclusiones y trabajo futuro i

Abstract ix

Acknowledgments xi

Contents xiii

List of Figures xvii

List of Tables xxi

List of Symbols xxiii

1 Introduction 1

1.1 A brief introduction to motion analysis . . . 1

1.1.1 Motion analysis application . . . 3

1.1.2 Motion analysis problems . . . 5

1.2 The motion estimation problem . . . 8

1.3 The motion segmentation problem . . . 9

1.4 Parametric motion models . . . 11

1.5 Overview of the Thesis . . . 13

1.5.1 Objectives . . . 14

1.5.2 Contributions . . . 16

1.5.3 Methodology . . . 17

1.5.4 Document organization . . . 18

2 GLS-based parametric motion estimation **21**
2.1 Introduction 22
2.2 A brief review on image registration techniques 26
2.3 Details on feature-based image registration 28
2.3.1 Step 1: Feature detectors 28
2.3.2 Step 2: Feature descriptors and matching 31
2.3.3 Step 3: Motion parameters estimation 34
2.3.4 Feature-based image registration algorithm 37
2.4 GLS-based motion estimation 38
2.4.1 GLS for general problems 38
2.4.2 An example of using the GLS for plane estimation 42
2.4.3 A GLS-based model in motion estimation problems 42
2.4.4 The role of the weights in GLS-based motion estimation . . 47
2.5 Two-steps image registration algorithm 51
2.6 Experiments and results 52
2.6.1 Experiments using the affine motion model 52
2.6.2 Experiments using the projective motion model 60
2.7 conclusions 61

3 Global mot. est. under non-uniform illumination **65**
3.1 Introduction 66
3.2 Invariant color representations 67
3.3 Spatially varying illumination model 72
3.4 GLS-based global motion estimation under varying illumination . . 73
3.4.1 Motion and illumination parameters initialization 74
3.5 Experimental results 75
3.6 Conclusions 80

4 Quasi simultaneous motion est. and seg. **81**
4.1 Introduction 82
4.2 Terminology used in this chapter 84
4.3 Quasi-simultaneous parametric multimodel fitting 85
4.4 The proposed algorithm 87
4.4.1 Motion parameters estimation for a model 87
4.4.2 Algorithm outline 89
4.4.3 Inliers/Outliers region classification 92
4.4.4 An illustrative example of the behavior of the algorithm . . 93
4.4.5 Refining segmentation 98
4.5 Experimental results 99

4.6 Conclusions 102

5 Contributions, conclusions and future work 105

5.1 Summary of contributions and main conclusions 105

5.2 Future Work 108

5.3 Publications 108

A Test images 113

B Image registration results 123

Bibliography 133

List of Figures

1.1 *Hal 9000* guesses the conversation between Dave and Frank reading their lips. 2
1.2 Panoramic image mosaicing example 4
1.3 The creation of *Gollum* by motion analysis techniques 6
1.4 An example of a video-game using computer vision techniques . . . 7
1.5 Image registration example 9
1.6 Optical flow calculation example 10
1.7 Motion segmentation example 11
1.8 Affine motion model transformations 14
1.9 An example of the possible transformations that can be obtained using the projective motion model. 15

2.1 Mutual information image registration 27
2.2 An example of a peak in the discrete cross correlated function. . . 28
2.3 Invariant feature detector 30
2.4 SIFT example 32
2.5 Matching feature points 34
2.6 Feature-based (SIFT-based) image registration algorithm 37
2.7 Real example of SIFT performance 38
2.8 Real example of the performance of the matching procedure 39
2.9 Example of the two types of residuals for a well known line fitting regression problem 40
2.10 GLS-based plane estimation example 43
2.11 Complete generalized least squares motion estimation algorithm . . 47
2.12 Simplified generalized least squares motion estimation algorithm . 48
2.13 The Role of the weights in GLS-bases Motion Estimation 50

2.14 Sample images from *Boat*, *Bark*, *Ubc*, *Bikes* and *Leuven* image sets 56
2.15 Accuracy results 57
2.16 Percentage of outliers introduced at each set. 58
2.17 Accuracy results when an image patch of 200 × 200 has been introduced as a source of outliers 59
2.18 Sample images from *graf* set 61
2.19 Registration results for images from *graf* set. 62
2.20 Registration results for images from *graf* set. 63
2.21 Input satellite images. 63
2.22 Satellite image registration results. 64

3.1 Image registration under non-uniform illumination changes. 68
3.2 An example of how invariant representation can help to image processing techniques 69
3.3 Multipliers used in this work and examples 76
3.4 Results obtained when the images I_2 are used as second image . . 77
3.5 Results obtained when the images $I_2^{Gd}$are used as second image. Black: BCA-based; White: DIM-based. 78
3.6 Results obtained when the images I_2^{Gn} are used as second image. Black: BCA-based; White: DIM-based. 78
3.7 Mosaic image created using as input images from the Boat set. . . 79
3.8 Input image, estimated one and grey level difference 80

4.1 Danuser and Stricker's multiple model extraction algorithm. 86
4.2 Danuser and Stricker's single model extraction algorithm. 88
4.3 Our proposed quasi-simultaneous motion estimation and segmentation algorithm. 89
4.4 Two examples of initial models. 90
4.5 Inliers/Outliers region classification algorithm. 93
4.6 Algorithm to include new neighbor regions to a model. 94
4.7 Synthetic sequence 95
4.8 Hypothetic results of the segmentation process and labels of each region 96
4.9 Adjacency Graph 96
4.10 Two possible initial models 97
4.11 Classification step 97
4.12 Interchange of regions between model 98
4.13 Both images of the synthetic sequence and results 100
4.14 Both images of the real sequence and results 101

4.15 Optic flow computed from results of the synthetic and real sequence 103
4.16 Refined segmentation for test sequences 104

5.1 Relation between publications arisen so far from this PhD. 111

A.1 Images from *Boat* image set 114
A.2 Images from *Bark*image set 115
A.3 Images from *Bikes* image set 116
A.4 Images from *Ubc* image set 117
A.5 Images from *Leuven* image set 118
A.6 Images from *Graf* image set 119
A.7 Images from *Tree* image set 120
A.8 Images from *Wall* image set 121

B.1 Registration results from *Boat* image set 124
B.2 Registration results from *Boat* image set 124
B.3 Registration results from *Bark* image set 125
B.4 Registration results from *Bark* image set 125
B.5 Registration results from *Bikes* image set 126
B.6 Registration results from *Ubc* image set 126
B.7 Registration results from *Leuven* image set 127
B.8 Registration results from *Tree* image set 127
B.9 Registration results from *Graf* image set 128
B.10 Registration results from *Graf* image set 128
B.11 Registration results from *Graf* image set 129
B.12 Registration results from *Graf* image set 129
B.13 Registration results from *Wall* image set 130
B.14 Registration results from *Wall* image set 130
B.15 Registration results from *Wall* image set 131

List of Tables

1.1 Main properties of parametric motion models 12

3.1 Resume of some of the main color invariants 71

List of Symbols

Images

I_1	First image of a sequence, also called **test** image.
I_2	Second image of a sequence, also called **reference** image.
$I_1(x_i, y_i)$	Grey level of test Image at point (x_i, y_i).
$I_2(x'_i, y'_i)$	Grey level of reference Image at point (x'_i, y'_i).
(x_i, y_i)	Column and row of pixel related to observation λ_i.
(x'_i, y'_i)	Column and row of transformed (by motion parameters) pixel.
$I_1^x(x_i, y_i)$	Column gradient of Image I_1 at point (x_i, y_i).
$I_1^y(x_i, y_i)$	Row gradient of Image I_1 at point (x_i, y_i).
$I_2^x(x'_i, y'_i)$	Column gradient of Image I_2 at point (x'_i, y'_i).
$I_2^y(x'_i, y'_i)$	Row gradient of Image I_2 at point (x'_i, y'_i).

Motion Models

Translational

c_1	Horizontal offset.
c_2	Vertical offset.

Affine

a_1	Controls scaling and rotations in the horizonal direction.
b_1	Controls scaling and rotations in the horizonal direction.
a_2	Controls scaling and rotations in the vertical direction.
b_2	Controls scaling and rotations in the vertical direction.
α	Rotation degree.
K_x	Scale factor in the horizontal direction.
K_y	Scale factor in the vertical direction.
Sh_x	Shear factor in the horizontal direction.
Sh_y	Shear factor in the vertical direction.

Projective

d	Controls viewpoint changes.
e	Controls viewpoint changes.

Generalized Least Squares

Θ	Objective function to be minimized.
Θ_ν	Objective function based on the residuals of the observations.
Θ_ϵ	Objective function based on the residuals of the functions.
υ	Residual of observations, $\upsilon = \lambda - \tilde{\lambda}$.
ϵ	Residual of functions.
χ	Vector of p parameters to be estimated,$\chi = (\chi^1, \ldots, \chi^p)^t$.
λ	Unperturbed vector of r observations λ_i, $\lambda = (\lambda_1, \ldots, \lambda_r)^t$.
λ_i	Observation with n components $\lambda_i = (\lambda_i^1, \ldots, \lambda_i^n)$
$\tilde{\lambda}$	Actually measured vector of observations.

$F(\chi, \lambda)$	Restrictions of the minimization problem. $F(\chi, \lambda) = (F_1(\chi, \lambda_1), \dots, F_r(\chi, \lambda_r))^t$.
$F_i(\chi, \lambda_i)$	Set of f functions $F_i(\chi, \lambda_i) = (F_i^1(\chi, \lambda_i), \dots, F_i^f(\chi, \lambda_i))^t$.
r	Dimension of the vector of observations λ.
p	Dimension of the vector of parameters χ.
n	Dimension of the observation vector λ_i.
f	Dimension of the set of functions $F_i(\chi, \lambda_i)$.
j	Iteration number.
i	Number of observation $i = 1 \dots r$.
A	$A = \partial F / \partial \chi$.
B	$B = \partial F / \partial \lambda$.
E	$E = -F(\widehat{\chi}(j), \widehat{\lambda}(j))$.
Q	$Q = (BB^t)^{-1}$.

Feature-based Image Registration

Ψ_1	Set of feature points detected at Image 1.
Ψ_2	Set of feature points detected at Image 2.
Π_1	Set of descriptors of the Image 1.
Π_2	Set of descriptors of the Image 2.
n_1	Number of feature points detected at Image 1.
n_2	Number of feature points detected at Image 2.
(x_i, y_i)	Localization of the i-th feature point detected.
s_i	The scale of the i-th feature point detected.
α_i	The orientation of the i-th feature point detected.

Quasi simultaneous motion estimation and segmentation

I_1	First image in the sequence.
I_2	Second image in the sequence.
M	A model which consist in a vector of parameters χ and a set of observation that supports the model Φ. $M = [\chi, \Phi]$.
χ_i	Vector of parameters of i-th model.
Φ_i	Set of observation that supports the i-th model.
Υ	Set of extracted models. $\Upsilon = [M_1, M_2, \ldots, M_n]$
Ω	Set of not yet classified observations.
$\Re$	Set of input observations.
Γ	Set where the observations with problems are saved.
G	Adjacency graph.
j	Iteration

Chapter 1

Introduction

Contents

1.1 A brief introduction to motion analysis 1
1.2 The motion estimation problem 8
1.3 The motion segmentation problem 9
1.4 Parametric motion models 11
1.5 Overview of the Thesis 13

IN this first chapter, some of the main concepts about this thesis will be introduced. First, a general introduction to motion analysis will be done, since motion analysis is the principal computer vision area where this work can be classified in. After that, both, the motion estimation and motion segmentation problems will be commented. These problems are the ones addressed in this work. In the last part of the chapter, an overview of the thesis will be done, describing the aims, contributions and the organization of the thesis.

1.1 A brief introduction to motion analysis

Computer vision -also called *Artificial Vision*- has been extensively used in the science and fiction literature and in the cinematography for the last decades. Perhaps, the most famous computer of the history having an artificial vision system is the computer of *Arthur C. Clarke* novel and *Stanley Kubrick* film *"2001 An Space Odyssey"* [Kubrick and Clarke, 1968]. It was called *HAL 9000.* Among other visual capabilities, that computer could, in one of the most famous scene of the film, read the lips of two astronauts (see Figure 1.1).

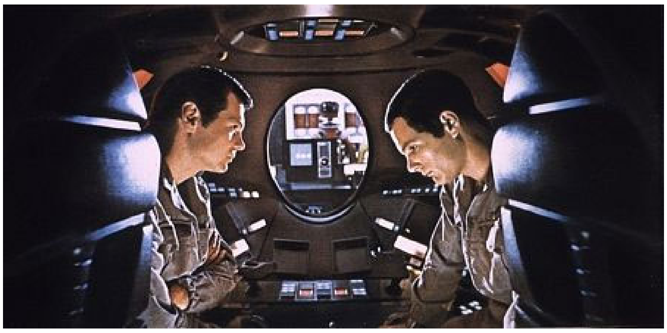


Figure 1.1: Hal 9000 *guesses the conversation between Dave and Frank reading their lips.*

In 1968 -the year that the film was filmed- the scriptwriters thought that many of the abilities of *HAL 9000*, and also many other computer vision-related applications viewed in the film, might be solved by the year 2001. They got expert advice from some of the most important scientists of that period. At that time, artificial vision was conceived as a mere imitation of human vision. It was assumed that it would be relatively easy to make artificial mechanisms and algorithms able to carry out whatever visual operation than human can do. Nowadays, we are in 2008 and *HAL 9000* continues being science fiction, mainly due that many aspects of human vision have not been completely understood yet. However, for the last decades, it has been invested an important effort in computer vision. In addition, on the one hand, processors speed and hard disk capacity have been hugely improved, and on the other hand, the prices of hardware have been interestingly reduced. These facts have produced an important progress in the challenge to obtain an artificial vision system as good as the human one.

Motion analysis is one of the most important research fields in computer vision, since real world is in motion and it is obvious that much more information can be obtained from a moving scene than from a static one. Therefore, the study and the analysis of motion is a fundamental task to understand the world where we live in. In addition, it will be a principal requirement to whatever machine or organism interacting with its environment.

One of the main aims of motion analysis is to implement an artificial motion perception system similar to the human one. Human motion perception can be defined as the process of inferring the speed and direction of elements in a scene based on visual input. Although this process could seem straightforward to most observers, it has proven to be a difficult problem from a computational perspective, and extraordinarily difficult to explain in terms of neural processing.

1.1.1 Motion analysis application

In spite of the difficulties commented before, motion analysis has nowadays many applications. Some of the most important applications of motion analysis are the followings:

- **Panoramic image mosaicing**: One of the most popular application is the creation of panoramic image mosaics [Brown and Lowe, 2003], [Szeliski, 2004]. This task is very related to the problem known as Image Registration [Brown, 1992], [Zitova and Flusser, 2003]. We refer to Image Registration as the process of finding the correspondence between a set of pixels in one image with a set of pixels in a second image, where both images are acquired from the same scene but may be captured at different time, using different sensors and having different viewpoints. Image Registration will be studied in Chapter 2. The creation of panoramic images consists of estimating the deformation between a set of images with respect to one base image. Then, all the images are merged into a new one, called as mosaic. An example of this application is showed in Figure 1.2 where the images where taken from `http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html` and the panorama image was created by Brown and Lowe's algorithm [Brown and Lowe, 2003].

- **Traffic monitoring**: Traffic monitoring [Ferrier et al., 1994], [Setchell, 1997], [Kastrinaki et al., 2003], [Tai et al., 2004], [Rad and Jamzad, 2005], [Ji et al., 2006] is one of the challenging problems in computer vision in general and in motion analysis in particular. Traffic monitoring involves the collection of data describing the characteristics of vehicles and their movement through road networks. Vehicle counts, vehicle speed, vehicle path, vehicle density, vehicle length, weight, class (car, van, bus) and vehicle identity via the number plate are all examples of useful data. Such data may be used for one of the following purposes [Setchell, 1997]:

  - Law enforcement: Speeding vehicles, dangerous driving, illegal use of bus lanes, detection of stolen or wanted vehicles.
  - Automatic toll gates: Manual toll gates require the vehicle to stop and the driver to pay an appropriate rate. In an automatic system the vehicle would no longer need to stop. As it passes through the toll gate, it would be automatically classified in order to calculate the correct rate. The vehicle's number-plate would be automatically deciphered and a monthly bill would be sent to the owner.

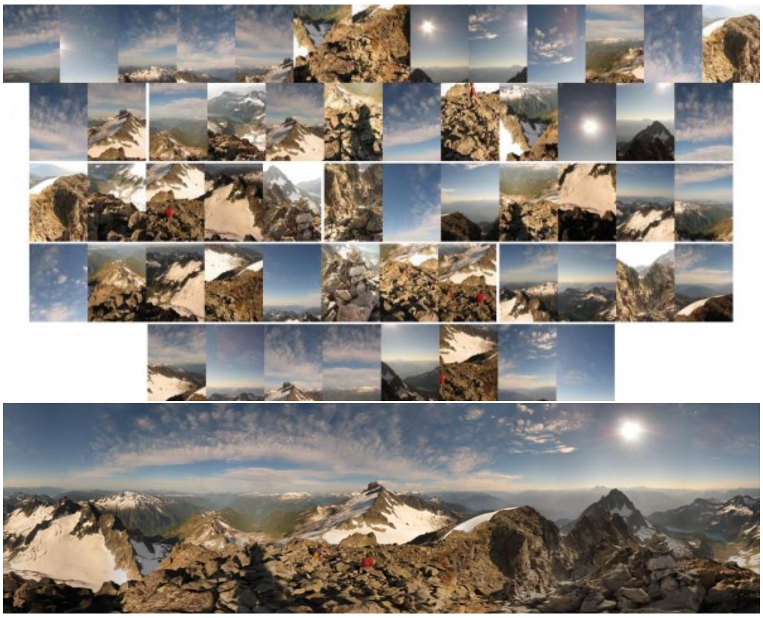


Figure 1.2: *Panoramic Image mosaicing example. Top: input images from a scene. Below: the resulting mosaic image.*

  - Congestion and Incident detection: Traffic queues, accidents and slow vehicles are potentially hazardous to approaching vehicles. If such incidents can be detected then variable message signs and speed limits can be set up-stream in order to warn approaching drivers.
  - Increasing road capacity: Increasing the capacity of existing roads is an attractive alternative to building new roads. Given sufficient information about the status of a road network, it is possible to automatically route traffic along the least congested roads at a controlled speed in order to optimize the overall capacity of the network.

- **Surveillance**: From the events on September 11th in New York and more recently in Madrid and London, surveillance has become one of the most important research fields for governments of many countries. Thus, visual surveillance in dynamic scenes, especially for humans and vehicles, is currently one of the most active research topics in computer vision [Hu et al., 2004], [Prati et al., 2003], [Radke et al., 2005], [Cucchiara et al., 2003], [Wang et al., 2003], [Haritaoglu et al., 2000]. There is a wide spectrum of promising applications, including access control in special areas, human identification at a distance, detection of anomalous behaviors, detection of suspicious abandoned objects, interactive surveillance using multiple cameras and so on.

- **Video-conference**: Complex algorithms have been developed in order to compress images to speed up the transmission ([Tseng, 2004], [Grecos et al., 2004]). Tracking algorithms have also been developed to put the focus on the speaker in a conference. This is a hot subject today due to the potential applications for mobiles phones.

- **Entertainment industry**: Motion analysis techniques are also widely used in television, video-games, cinema and other entertainment related industries [Ren et al., 2005], [Tu et al., 2007], [Kang et al., 2004]. For instance, *"The lord of the rings: The return of the king"*, winner of 11 academic awards (including best visual effects), is a good example of a film using motion analysis techniques. In this film, motion capture algorithms have been used to help graphic designers to create the digital character of *Gollum* [Serkis, 2003] (see Figure 1.3).

  Many television channels employ virtual environments. While the presenter is moving on a empty tv scene, the camera follows his movements and an algorithm mix the images captured by the camera with a virtual environment, allowing TV viewers to figure out that the presenter is into the virtual scene. Whether information is a good example of this application.

  A motion capture techniques are also used in many video-games to create digital characters with very similar performance than human ones, that is very useful in sport games. More recently, a new concept of gaming has been developed: it is a small video camera located on the top of the tv and plugged into a video console. The motion sensitive camera films you as you stand in front of the tv, putting your image on screen in the middle of the action. Figure 1.4 shows an screenshot of this game.

The previous list of applications is only a sample. There are many other interesting applications of motion analysis in real live problems.

1.1.2 Motion analysis problems

Motion analysis, such as many computer vision tasks is not free of difficulties. The most significative are the followings:

Figure 1.3: *To create the animation sequences, the team of the film used a combination of motion-capture from Andy Serkis' (left image) lively and expressive face, as well as traditional key-frame animation. Serkis, in addition to being the voice of Gollum, was also his physical presence on the film set. The films' actors interacted with Serkis, and then the animation team went through these scenes frame by frame and effectively painted Serkis out of the scene, and animated the Gollum model into it. The end-results seen in the films speak for themselves, and the team of the film won a Visual Effects Society Award for their work.*

- **The aperture problem**: This problem arises when there is not sufficient image intensity variation in the region of the image where motion is going to be estimated. In these situations, there are more than one possible motion that match the region. Only when there is enough image intensity variation in the region, for instance, in the edges of the objects of the image, the motion can be estimated with high accuracy.

- **Images are 2D**: Real word is 3D, but images are only 2D. The camera makes a transformation of the real 3D points of a scene in image 2D points. Therefore, information about the depth of the objects in the scene is completely lost. For instance, problems arise when several 3D points are projected in the same 2D point in the image, or when the real motion observed in the scene 3D is not observed at the scene 2D. The classical example is a gray sphere rotating in the world 3D. This rotational motion can not be seen in the image 2D. However, if the sphere is static but it is illuminated with a moving light, then in the projected scene the sphere seams it is moving.

- **Occlusions**: Occlusions arise when an object in a scene is covering some part of other object. This fact produces that motion analysis techniques could get confused, specially in tracking applications.

- **Limits of the motion model used**: The mathematic model used to explain the motion determines the information that we can extract. For

Figure 1.4: *An example of a video-game using computer vision techniques. A motion sensitive camera films you as you stand in front of the tv, putting your image on screen in the middle of the action.*

instance, if a translational model is used, reliable information can not be obtained if in the scene there are rotational motions, since not all the pixels of that area have the same translational motion. In the same way, if an affine motion model is applied, problems arise if there are objects in the scene at different deeps, since affine model supposes that all the objects in the scene are in the same plane.

- **More than one motion**: If in an area where we are assuming that there is only one motion, and in fact, there are several ones, then the different motions estimated will be contaminated for all the motions present in that area. Therefore, it can not be accurately estimated.

- **Outliers**: In statistics, classical methods rely heavily on assumptions which are often not put into practice. In particular, it is often assumed that the data are normally distributed. Unfortunately, when there are outliers in the data (i.e. data points with an extreme deviation from the mean), classical methods often have very poor performance. For instance, when using a least squares estimation, even the presence of a single outlier can affect the estimation of the model. In motion analysis all the pixels which do not follow the model of the main motion present in the scene could be considered as outliers. Robust statistics [Hampel et al., 1986], [Huber, 1981], [P.J. and A.M., 1987], [Ricardo A. Maronna, 2006] is the science into statistics whose main aim is to provide methods that emulate classical methods, but which are not unduly affected by outliers or other small deviations from model assumptions.

1.2 The motion estimation problem

When we talk about motion estimation, its is convenient to make a difference between global and local motion estimation. On the one hand, global motion estimation can be defined as the process that obtains a mathematical model that explains the deformation between two consecutive images from a sequence. This problem is also known as image registration ([Brown, 1992], [Zitova and Flusser, 2003]), which is one of the main topics of this thesis. Chapter 2 is completely devoted to the image registration problem. On the other hand, local motion estimation is the process of determining the displacements or velocities of pixels from one frame to another. This problem is also known as optical flow calculation ([Barron et al., 1994], [Beauchemin and Barron, 1995]). The main difference between the global and the local motion estimation problem lies in the size of the object where the motion have to be estimated. In global estimation, the object is the entire image. In local estimation the object can be a single pixel.

In order to best explain the difference between both definitions of motion estimation, the Figures 1.5 and 1.6 show one example of each problem. In Figure 1.5, a typical image registration problem is showed. The input are two satellite images from the same scene but they have been captured at different times. In this case, the motion is global, since the most of the pixel support the dominant motion model present at scene. The motion estimation algorithm must to be able to estimate the best suitable motion model that can explain the deformation between the two images. Once it has been estimated, it is posible to create a mosaic image showing the results of the estimation, i.e. we know how to transform the second image of the sequence to match with the first one. To explain the motion between the images, a parametric motion model can be used. This mathematical model can be used to calculate how the coordinates of a pixel from the reference image is moved to the coordinates at the target image. The most commonly used parametric motion models are described at Section 1.4

Figure 1.6 shows an example of optical flow calculation. Here, the aim is to estimate the displacement of each pixel. The displacement is showed using an arrow. The reader is referenced to [Barron et al., 1994] for a comprehensive study of some of the most popular optical flow estimation techniques.

In this work, we mainly have worked on global motion estimators for image registration. Therefore, hereafter in this document, the terms image registration and global motion estimation will be used indistinctively. The aim is to develop

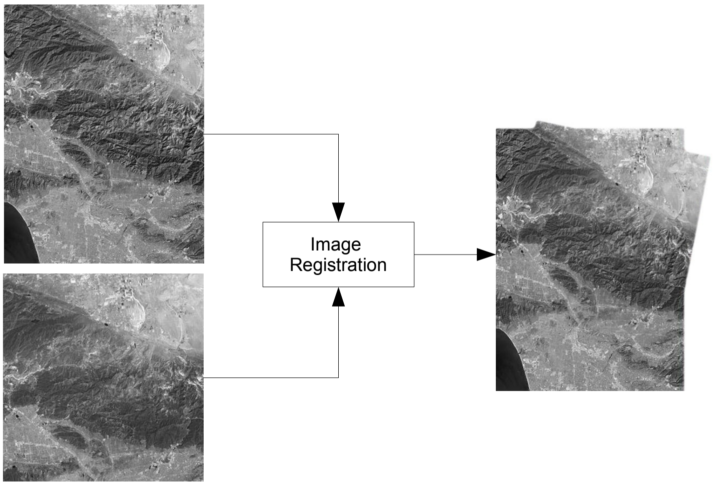


Figure 1.5: *Image Registration example: Two images from the same scene are the input. The result of the registration process is a mosaic with the two images merged.*

an image registration algorithm able to deal with large deformations while achieving high accuracy in the parameters estimation. Chapter 2 explains the proposed global motion estimation algorithm, which can deal with any translation, any rotation degree, very strong scale changes, blur, jpg compression, moderate viewpoint changes and global illumination changes. The database of images that have been used in our experiments are showed in appendix A. Chapter 3 introduces a dynamic image model to help to the proposed global motion estimator to deal with spatially varying illumination changes.

1.3 The motion segmentation problem

Motion segmentation consists of grouping together all the pixels in a image with the same apparent motion. It is a similar process to the segmentation of static images, but in this case, the pixels are grouped together following a similarity criteria based on motion, and not based on pixel colors. Figure 1.7 shows and example of Motion Segmentation. Two images are the input, and an image where each group of pixels with the same motion have been labelled with a different color, is the output. In this case, there are three groups, the background, that is static, and the two moving trucks. Another difference between motion segmentation and traditional color segmentation is that in the first case, it is

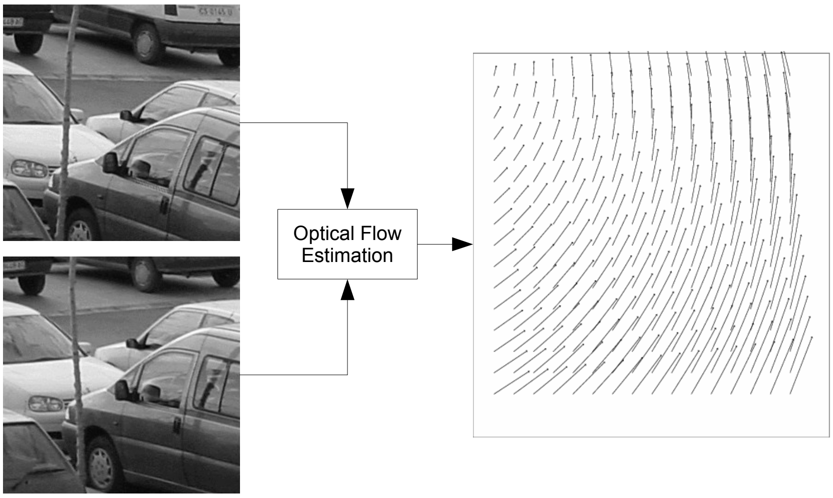


Figure 1.6: *Optical flow calculation example: Two images from the same scene are the input. The result of the estimation process is a set of displacement vectors showing the displacement of each pixel.*

necessary to have more than one image to perform the segmentation. At least two images are needed to observe motion.

Once the image sequence has been segmented according to the motion, it is posible to estimate the motion of each object. For instance, in the case showed at Figure 1.7 three motion models can be obtained. The first one related to the background and the other two related to each truck.

The problem of simultaneously estimating the motion while performing the segmentation is known as the motion estimation and segmentation problem. Performing motion estimation and motion segmentation simultaneously usually falls in a *Hen-and-egg* problem. It is due to the fact that data classification and parameter estimation strongly depend on each other. It is known that, on the one hand, if the data is well-classified, i.e, we know which pixel support which model, then it is easy to obtain accurate estimates for the parameters. On the other hand, if we know accurate estimates of the parameters, then it is straightforward to classify the pixels into the models.

Chapter 4 deals with the motion estimation and segmentation problem. We present a new approach that uses as motion estimator an algorithm based on some of the ideas proposed in Chapter 2. The proposed method accurately estimates the motion parameters while classifies the pixels into the motion models present

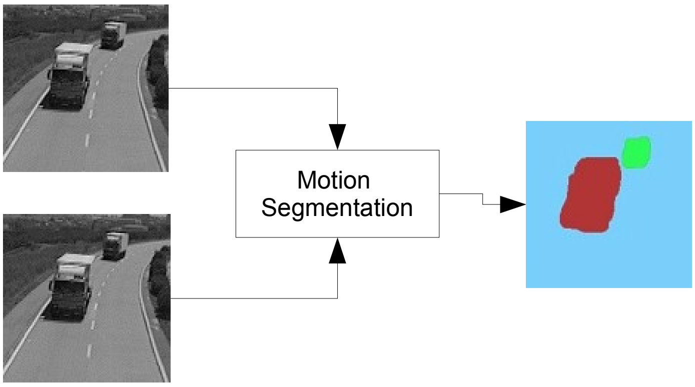


Figure 1.7: *Motion Segmentation example.*

in two consecutive frames.

1.4 Parametric motion models

In this section, the most used parametric motion models for motion estimation are presented. Parametric motion models are employed to calculate how the coordinates of a pixel from the reference image will be moved to the coordinates at the target image in motion estimation problems. A hierarchical classification is presented since each model is an extension of the previous one. The most common ones are the translational (with two parameters), the affine (with six parameters) and the projective (with eight parameters). Table 1.1 summarizes the main properties of the three motion models. Note that the translational motion model can only deal with translations, the affine motion model can deal with translations, rotations, scale changes and shear. Finally, the projective motion model can deal with all the previous ones and also with viewpoint changes.

The motion models, in increasing order of complexity, are the following, with χ being the vector of motion parameters of each motion model, (x_i, y_i) the coordinates of a point and (x'_i, y'_i) the transformed coordinates of that point:

- **Translational**: $\chi = \{c_1, c_2\}$, where c_1 and c_2 are the horizontal and vertical offsets, respectively. It is defined as follows:

$$\begin{cases} x'_i = x_i + c_1 \\ y'_i = y_i + c_2 \end{cases} \tag{1.1}$$

	Translational	Affine	Projective
Num. parameters	2	6	8
Parameters	$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$	$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix}$	$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ d & e & \end{bmatrix}$
Translation	$\surd$	$\surd$	$\surd$
Rotation	×	$\surd$	$\surd$
Change of scale	×	$\surd$	$\surd$
Shear	×	$\surd$	$\surd$
Viewpoint changes	×	×	$\surd$

Table 1.1: *Main properties of parametric motion models.* $\surd$ *denotes that the motion model can deal with, and* × *the opposite.*

- **Affine**: $\chi = \{a_1, b_1, c_1, a_2, b_2, c_2\}$, where c_1 and c_2 are the horizontal and vertical offsets, respectively. The parameters a_1, a_2, b_1 y b_2 are used to control the magnitude of change of scale, rotations and shear transformations. It is expressed as follows:

$$\begin{cases} x_i' = a_1 x_i + b_1 y_i + c_1 \\ y_i' = a_2 x_i + b_2 y_i + c_2 \end{cases} \tag{1.2}$$

If α is the angle of rotation, K_x and K_y the scale factors and Sh_x and Sh_y the shear factors, then Equation 1.2 can be written as follows:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} K_x & 0 \\ 0 & K_y \end{pmatrix} \begin{pmatrix} 1 & Sh_y \\ Sh_x & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{1.3}$$

Thus, the affine parameters a_1, b_1, a_2 and b_2 can be calculated as follows:

$$\begin{cases} a_1 = K_x \cos\alpha - K_y Sh_x \sin\alpha \\ b_1 = K_x Sh_y \cos\alpha - k_y \sin\alpha \\ a_2 = K_x \sin\alpha + K_y Sh_x \cos\alpha \\ b_2 = K_x Sh_y \sin\alpha + k_y \cos\alpha \end{cases} \tag{1.4}$$

An affine transform maps a triangle into a triangle and a rectangle into a parallelogram.

- **Projective**: Using affine motion, problems can arise when there is a strong viewpoint change between images. In order to cope with viewpoint changes, projective motion can be used instead. In this case, the vector of parameters is expressed as: $\chi = \{a_1, b_1, c_1, a_2, b_2, c_2, d, e\}$. The projective parametric motion model is defined as follows:

$$\begin{cases} x'_i = \dfrac{a_1 x_i + b_1 y_i + c_1}{d x_i + e y_i + 1} \\ y'_i = \dfrac{a_2 x_i + b_2 y_i + c_2}{d x_i + e y_i + 1} \end{cases} \tag{1.5}$$

  In contrast to the affine transform, the projective transform is nonlinear. It maps lines into lines but only lines parallel to the projection plane remain parallel. A rectangle is mapped into an arbitrary quadrilateral.

Figure 1.8 shows and example of how the affine motion model can transform a square. The original square is the blue one and the transformed is the red one. From top to bottom, left to right, the first graphic shows a translation of two pixels in both directions (i.e. $c_1 = c_2 = 2$). The second one shows a rotation of $\alpha = 30$ degrees. The third one illustrates a change of scale of factor $Kx = Ky = 0.75$. Finally the last graphic shows the effects of a shear of factor $Sh_x = Sh_y = 0.5$. The center of all transformations is located at the center of the blue square.

Figure 1.9 shows four examples of how an square can be transformed by using the projective motion model. From top to bottom, left to right, the first graphic shows a projective transformation where the parameters d and e are $d = 0.1$ and $e = 0.0$. The second one, $d = 0.0$ and $e = 0.1$. The third one, both d and e have been set to 0.1. Finally, the last graphic illustrate a projective motion where both d and e parameters have been ser to -0.1.

1.5 Overview of the Thesis

The work of this thesis has been developed in the framework of several research projects that have been carried out in the Computer Vision group at Jaume I University and supported by public funds. They are the followings:

- Projects: *GV97-TI-05-27* and *CTIDIB/2002/333*, from the Conselleria de Educació Cultura i Ciència, Generalitat Valenciana.

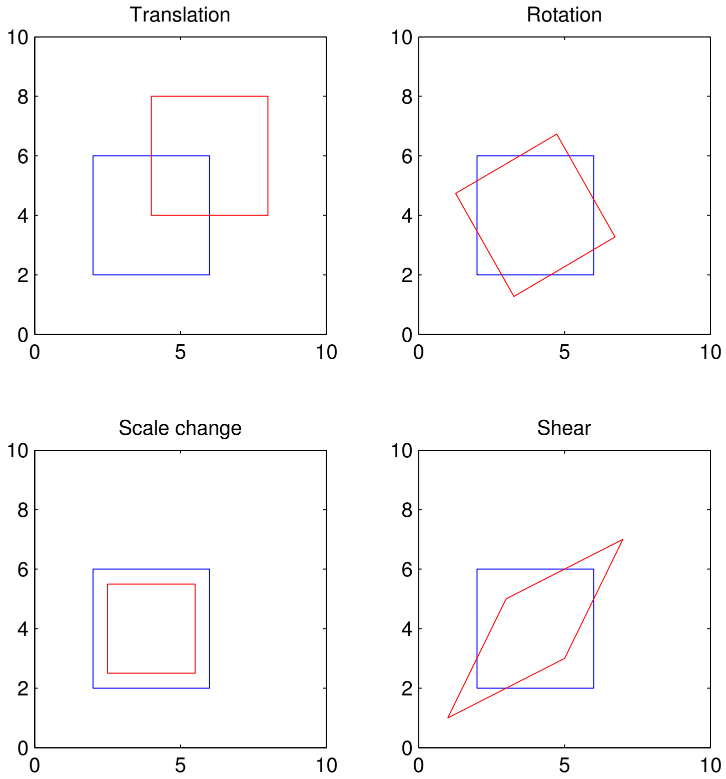


Figure 1.8: *The affine motion model can deal with translation (top-left), rotations (top-right), scale changes (bottom-left) and shear(bottom-right).*

- Projects: *TIC98-0677-C02-01*, *TIC 2001-4570-E*, *DPI2001-2956-C02-02* and *ESP2005-07724-C05-05* from Spanish Ministerio de Educación y Cultura.
- Project *IST-2001-37306* from European Union.

As it was pointed out, this thesis mainly deals with motion estimation for image registration and motion segmentation problems. In this section the main objectives, the methodology, the main contributions and the organization of the contents of this thesis are explained.

1.5.1 Objectives

The general objectives of this thesis are:

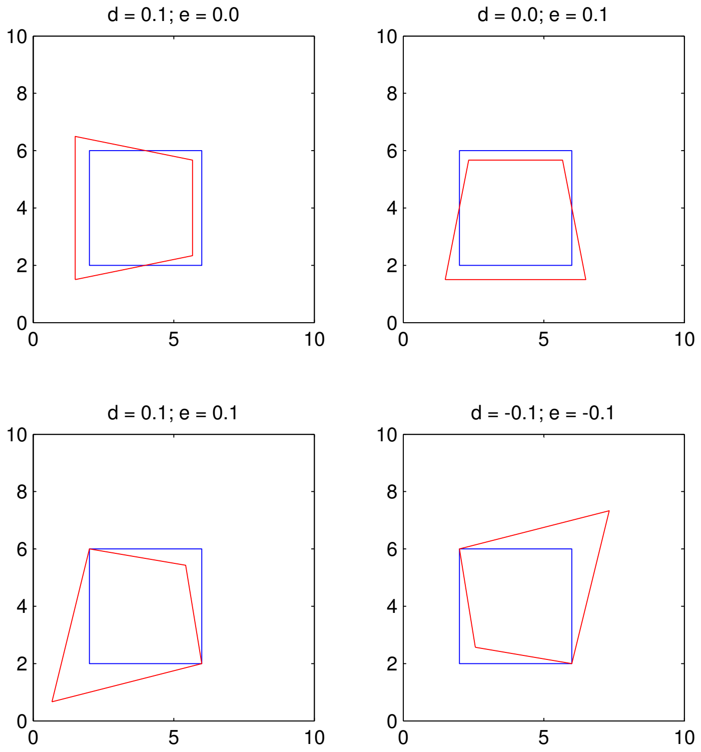


Figure 1.9: *An example of the possible transformations that can be obtained using the projective motion model.*

- Study the properties of the main motion estimation algorithms, paying special attention to the ones that are used in image registration and motion segmentation applications and also to the principal techniques to deal with outliers in the data set.

- Design an accurate and tolerant to outliers motion estimator to be used in image registration and motion segmentation applications.

- Design a new, accurate, tolerant to outliers and able to deal with large motion and with non-uniform illumination changes image registration technique.

- Study the main properties of the most important techniques to solve the motion segmentation problem.

- Design a new motion segmentation technique able to accurately extract the different motions present in an image sequence.

The assumptions and conditions for the work carried out in this thesis are the following:

- Although only gray level images are used for the estimation process, the proposed image registration technique developed is also applied to color images. That is, when the input are color images, first the images are converted to gray level and then the motion parameters are estimated. Once the parameters have been estimated, a resulting mosaic image can be created using the original color images. From our experience, gray level images is enough to accurately estimate the motion parameters.

- Visual processing is exclusively 2D.

- Sequences of two images are used. Longer sequences may be considered in future developments.

- More than one moving object can appear in the sequence.

- The motion observed in the sequence can be produced by different factors, such as camera movements, moving object in the scenes, viewpoint changes, different illumination conditions between frames, etc.

- Images have been captured using a single camera. Although stereo vision is very related to motion estimation, this thesis does not deal with it.

- The affine and projective parametric motion models are used, but all the algorithms presented in this thesis can be also developed to use more complex motion models.

1.5.2 Contributions

Once known the goals described in the previous section, we now briefly summarize our contributions and achievements. More detail about conclusions and contributions will be given in Chapter 5.

- Regarding the motion estimation problem, we have reviewed the literature and studied a number of different techniques. Some of them have been used for comparison purposes.

- We have studied the GLS mathematical framework to be applied to motion estimation problems. In this sense, we have proposed a new GLS-based motion estimation technique to be applied to image registration and motion segmentation techniques.

- We have studied the problem of achieving large motion in image registration. We have reviewed some of the most important techniques, the majority of them related to the extraction of features invariant to rotations, scale changes, small viewpoint changes, etc.

- We have studied the problem of registering two images in the presence of non-uniform illumination changes.

- An accurate image registration technique able to deal with large motion and non-uniform illumination changes and tolerant to outliers has been designed and successfully tested.

- Regarding to the motion segmentation problem, we have also reviewed the literature and studied a number of different techniques.

- Finally, a new quasi simultaneously motion estimation and segmentation technique has been proposed.

1.5.3 Methodology

ANSI *C++* has been selected as programming language to implement the main part of the algorithms used in this thesis. Only a small part of the algorithms have been implemented using *Matlab*. In both cases, the source code has been written to fulfil the requirement that the code must to work on *Windows* and on *Linux* operating systems.

The election of *C++* is due to the fact that, nowadays, it has became the most common used programming language for computer vision researchers. Practically the whole tools and libraries related to computer vision have been written using the *C/C++* programming language. This fact allows researchers to easily share the code of theirs algorithms.

In last years, *Matlab* is gaining supporters because it is easier and faster to write prototype programs. But, its main disadvantage is that the program execution is much slower than the programs compiled using C++. *Matlab* is commonly used to create the first version of an idea, i.e. the prototype. When the developer confirms that the algorithm works well, then it can be implemented using *C++* to obtain a fast release version. *Matlab* is also very useful to create graphics to visualize data.

In windows, the Microsoft *Visual C++* programming environment has been used. In *Linux*, the ANSI GNU *g++* compiler has been used.

In order to test the different techniques found in the bibliography, we have used as long as it was possible, the original source code developed by the authors' papep. When it has not been possible to get the original code, we have tried to get in touch with the authors in order to solve the doubts about theirs works.

We have used the *PGM* (portable graymap) and *PPM* (portable pixmap) image file formats for gray scale and for color images, respectively, since they are easy to handle and are widely used for computer vision scientists. The images used for testing the developed algorithms have been obtained from three sources:

- Most of them from public databases. The Appendix A show some of them.
- Thanks to the collaborations of many authors who have sent them to us.
- By ourselves using digital cameras.

1.5.4 Document organization

This thesis has been organized in 5 chapters. Chapters 2 and 3 deal with the motion estimation problem applied to image registration, and chapter 4 deals with the motion segmentation problem. It would be more interesting to read the thesis following the numeric order of the chapters as they have been written, since some important concepts that will be used at chapter 4, are explained in previous ones. The contents of each chapter (excluding Chapter 1) are explained as follows:

- The core of this thesis lies in the **Chapter 2**, where the most important contents are explained. That is, our approach to solve the motion estimation problem applied to image registration. In this chapter, the proposed Generalized Least Squared (GLS) motion estimator is widely explained. A review of some of the most important image registration techniques are also commented. We pay special attention to the ones that have been chosen to be compared with our approach. Finally, in the last part of the chapter, a comparison among some of the most successfully techniques versus the proposed one is shown to illustrate the performance of the proposed method.

- **Chapter 3** deals with the problem of the estimation of the motion parameters under non uniform illumination changes. In particular, the proposed GLS-based algorithm has been adapted in order to allow to register images with non spacial uniform illumination changes.
- **Chapter 4** introduces the motion segmentation problem, making a brief review of the most important techniques found in the literature of computer vision and presenting our approach to solve the motion segmentation.
- **Chapter 5** presents the most important conclusions according to the aims, methods and results of this thesis. In addition, some ideas for future work are also discussed.

Furthermore, two appendixes have been done. The Appendix A shows the images used in the experiments. The Appendix B show several image registration results.

Chapter 2

Generalized least squares-based parametric motion estimation

Contents

2.1 Introduction 22

2.2 A brief review on image registration techniques . . . 26

2.3 Details on feature-based image registration 28

2.4 GLS-based motion estimation 38

2.5 Two-steps image registration algorithm 51

2.6 Experiments and results 52

2.7 conclusions 61

The estimation of parametric global motion has had a significant attention during the last two decades, but despite the great efforts invested, there are still open issues. The most important ones are related to the accuracy of the estimation and to the ability to recover large deformation between images.

In this chapter, a new generalized least squares-based motion estimator is proposed. The non-linear Brightness constancy assumption is directly used instead of using the classical approach by linearizing the minimization problem using the optical flow equation. In addition, the proposed formulation of the motion estimation problem provides an additional constraint that helps to match the pixels by using the image gradient in the matching process. That is achieved by means of a weight for each observation, assigning high weight values to the observations considered as inliers, i.e. the ones that support the motion model, and low values to the ones considered as outliers. The accuracy of our approach has been tested using challenging real images using both affine and projective motion models. Two Motion Estimator techniques that uses iteratively reweighted least squares-based (IRLS) techniques to deal with outliers, have been selected for comparison purposes. The results obtained show that the proposed motion estimator can obtain, in most cases, more accurate estimates that the IRLS-based techniques.

2.1 Introduction

Image registration [Brown, 1992] is a key problem in many applications of computer vision and image processing such as optical flow computation [Lucas and Kanade, 1981], [Bad-Hadiashar and Suter, 1998], medical imaging [D'Agostino et al., 2003], [Periaswamy et al., 2000], motion segmentation [Bad-Hadiashar et al., 2002], [Montoliu and Pla, 2005], [Odone et al., 2000], image mosaicing [Brown and Lowe, 2003], [Dufournaud et al., 2004], [Szeliski, 2004] among other. Despite the large amount of work in this area, there are still open issues mainly related to the accuracy of the estimation [Brox et al., 2004], [Nir et al., 2008], the convergence of the estimation algorithm [Keller and Averbuch, 2008], [Keller and Averbuch, 2004], [Baker and Maththews, 2004], [Baker and Matthews, 2002], [S. Baker and Ishikawa, 2003], the ability to recover large deformation between images [Brown and Lowe, 2003], [Dufournaud et al., 2004] and even to the ability to recover motion in the presence of illumination changes and shadows [Pizarro and Bartoli, 2007], [Kim et al., 2004], [Bartoli, 2006].

We refer to *Image Registration* as the process of finding the correspondence between a set of pixels in one image with a set of pixels in a second image, where both images are acquired from the same scene but may be captured at different time, using different sensors and having different viewpoints. Zitova and Flusser [Zitova and Flusser, 2003] divided image registration related applications into four groups, according to the image acquisition procedure:

- *Multiview analysis* where the images have been captured using different viewpoints.
- *Multitemporal analysis* where images from the same scene are captured at different times.
- *Multimodal analysis* where the images are captured using different sensors.
- *Scene to model registration* where images and a model of the scene are registered.

Some Image Registration related problems do not fall in any of the previous groups. More complex situations can occur, like in problems where time, viewpoint and the sensor change in a simultaneous way.

One of the most popular technique to deal with image registration is the use of optimization-based motion estimation methods, such as Least Squares (LS) regression techniques. Optimization methods, also known as direct methods, are based on estimating a vector of parameters that minimize (or maximize) an objective function. The main advantage of optimization-based methods is their accuracy because of the large volume of data implies that motion parameter estimation is heavily over-constrained, since a small number of parameters (6 for the affine motion model) are estimated using a large number of constraints.

In motion estimation problems, the objective function is usually based on the Brightness Constancy Assumption (BCA). The BCA is based on the principle of assuming that the changes in gray levels between the reference image and the test one are only due to motion. The main problem with BCA is that it is a non linear function. Therefore it has to be linearized in order to use a LS-based technique. The linear version of the BCA is known as optical flow equation and it has been widely used [Horn and Schunk, 1981; Barron et al., 1994]. In order to directly use the BCA instead of its linearized version, a non-linear estimator can be used, but then, the estimator usually becomes an iterative method, starting with an initial guess and updating the parameters at each iteration.

Iterative LS-based optimization methods for motion estimation problems have two important disadvantages. The first disadvantage is that they suffer from the presence of local minima and therefore the initial parameters used to initialize the method must not be very far from the solution in order to avoid falling into a local minimum. A well-know technique to cope with this initialization problem is to use hierarchical (coarse-to-fine) techniques [Bergen et al., 1992b], [Bergen et al., 1992a]. However, even using hierarchical techniques, optimization methods are not able to cope with very large motions.

An alternative to solve that problem is to use feature-based techniques as initialization. Feature-based techniques are usually carried out in three steps. The first step is the selection/extraction of image features. Next, each feature in one image is compared with potential corresponding features in the other image. A pair of features with similar attributes are accepted as matches and are called control points. Finally, the parameters of the best transformation which models the motion between the images are estimated using the control points obtained in the previous step. The main limitation of the feature-based methods is their high dependence on how the detection and extraction of features from the images are performed. This can affect the accuracy of the registration in the case of using interest point detectors with low repeatability rate. However, important advances have been made in the last years in this area. Some researchers have developed interest point detectors and descriptors invariant to large rotations, changes of scale, illumination changes and even partially invariant to affine changes. See [Mikolajczyk et al., 2005] and [Mikolajczyk and Schmid, 2005] for a comparative study of scale and affine invariant interest point detectors and local descriptors, respectively. Szeliski [Szeliski, 2004] maintains that if the features are well distributed over the image and the descriptors are reasonably designed for repeatability, enough correspondences to permit image registration can usually be found. This is the case when using the feature detectors and descriptors reported at [Mikolajczyk et al., 2005], [Mikolajczyk and Schmid, 2005], which allow to register images with large deformations. Brown and Lowe's algorithm [Brown and Lowe, 2003] is a good example of this fact.

Another important disadvantage of motion estimation is the presence of outliers, like other parametric estimation problems. Occlusions due to the motion, illumination changes, new objects in the scenes and sensor noise, are some of the sources of outliers. That can affect the accuracy of the estimation. In fact, when using an ordinary least squares method as estimator, the accuracy of the estimation can drastically be affected even in the presence of a single outlier. M-Estimators techniques are some of the robust techniques [Hampel et al., 1986], [Huber, 1981], [Black and Anandan, 1996] that have been frequently used in the past years in computer vision to deal with outliers [Bober and Kittler, 1994b], [Odobez and Bouthemy, 1995], [Ayer and Sawhney, 1995]. They are aimed at reducing the influence of outliers in the global estimation. M-Estimators techniques can be easily transformed in iterative reweighted least squares methods (IRLS), where, at each iteration, a weight for each observation is calculated, obtaining high weight values the observations that are considered as inliers and low weight values the ones considered as outliers. Those weights are calculated as a function of the residuals of the objective function. In practical applications, there is a high probability of having a moderate number of outliers. M-Estimator techniques has been usually added to optimization-based motion estimation methods to improve accuracy against outlier contamination, and therefore, to improve the quality of the results of real image registration.

In this chapter a new Generalized Least Squares-based (GLS) non linear motion estimation technique is proposed as an alternative method to M-Estimators and other robust techniques to deal with outliers. As it will be shown, it can obtain as accurate or better results as the M-Estimators methods. The proposed formulation of the motion estimation problem provides an additional constraint that helps the matching process using image gradient information, since it is well known that the areas with more information for motion estimation are the ones that have intensity variations like in the object edges of the image. Matching the pixels of these areas from the reference image to the test image is crucial for accurate motion estimation. Occlusions, illumination changes, etc. are areas where matching is not possible due to the fact that corresponding pixels in the other image might not exit. Similarly to the IRLS technique, the constraint that arises from the proposed formulation can be interpreted as a weight for each observation, providing high values to the weights of the observations considered as inliers, i.e. the ones that support the motion model, and low values to the ones considered as outliers. Strictly speaking using the statistics terminology, the proposed GLS-based motion estimation algorithm is not what it is called a *robust* method, since it does not fulfill one of the main properties that robust methods should meet: to have a bounded influence function. However, the results obtained show that the proposed method can deal with outliers, in terms of accuracy, like robust techniques can do.

In addition, to obtain an accurate image registration method able to cope with large deformations, a feature-based step is used to obtain the initial motion parameters, then the proposed GLS-based estimator is used to refine the parameters to obtain accurate estimates. At the first step, in order to cope with changes of scale, rotations, illumination changes and partially affine invariance, a SIFT-based technique [Lowe, 2004] has been used to detect and describe interest points, due to its excellent performance [Mikolajczyk and Schmid, 2005].

The main characteristics of the proposed method are summarized as follows:

- It uses a non-linear GLS-motion estimation technique. Therefore, the BCA can directly be used instead of its linearized version, the optical flow equation.

- To avoid falling in a local minimum, it uses a Feature-based method (SIFT-based) to obtain adequate initial motion parameters. Therefore, it can deal with large motion.

- The GLS-based motion estimation technique includes an additional constraint that helps to match the pixels using gradient information as a way to deal with outliers.

- Similarly to the IRLS technique, the constraint is expressed as a weight to each observation.

The rest of the chapter has been organized as follows: Section 2.2 presents a brief introduction on some of the main image registration techniques. Section

2.3 comments in detail the feature-based registration technique that has been used in our approach. The proposed GLS-based formulation of the motion estimation problem is described at Section 2.4. Section 2.5 explains the combined Feature-based with GLS-based image registration algorithm. In order to compare the proposed method with two well-known IRLS-based motion estimation techniques, Section 2.6 shows the experiments. Finally, the most important conclusions drawn from this paper are outlined in Section 2.7.

2.2 A brief review on image registration techniques

In the literature of computer vision and image processing we can find five main research directions on image registration: featured-based, optimization-based methods, mutual information-based, frequency domain-based and accumulative function-based. The first two are the ones that have been used in our approach. The last three are briefly resumen as follows:

- **Mutual Information-based**. The use of mutual information ([Pluim et al., 2003], [Rogelj et al., 2003], [D'Agostino et al., 2003]) is an alternative to feature-based method for multi-modal analysis, which is gaining supporters from its beginning in middle nineties. Mutual information-based registration algorithms have been mainly used in medical imaging to register CT-MR images, but little work has been done about registering other types of images. The concept of mutual information is based on the measure of information called entropy, which tries to asses the amount of information present in a signal. Preliminary works in multi-modal image registration proposed the use of the co-joint histogram of two images to be registered, as a feature space to find a solution for the registration problem. Figure 2.1 shows an example of two co-joint histograms calculated from two images. In the first case, both images are the same producing a straight line, this example simulates that the parameters of the transformation have been accurately estimated. In the second case, the second image vary from the first producing a spread line. In this case, the example simulates the effect of a non-accurate estimation.

- **Accumulative functions-based** The image registration methods based on voting/clustering algorithms, (for instance *RANSAC*, [Fischler and Bolles, 1981] and *Hough transform* [Illingworth and Kittler, 1988]), are robust against outliers, but their accuracy is ratter low, since they attempt to solve a problem defined in a continuous domain with a discrete solution

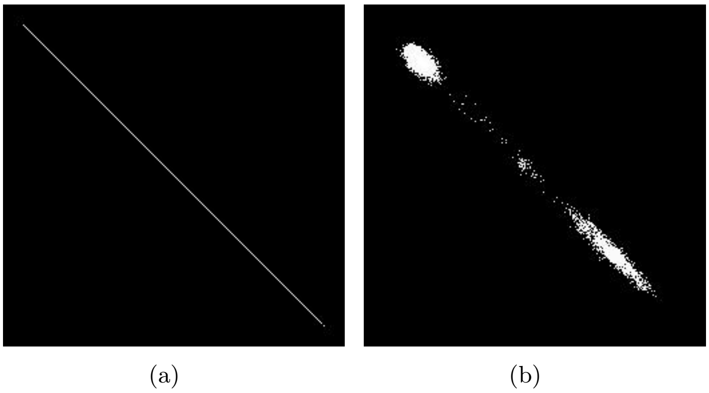


Figure 2.1: *Two examples of co-joint histograms (a) the parameters have been accurately estimated, (b) there are errors in the estimation.*

[Danuser and Stricker, 1998]. In addition, these methods need a considerable computational effort when the number of parameters increase like in the case of using affine and projective motion.

- **Frequency domain-based**. Frequency domain-based or phase correlation techniques have been also used in image registration [de Castro and Morandi, 1987], [Pearson et al., 1977], [Pla and Bober, 1997], [Lucceche et al., 1997]. These techniques estimate the relative shift between two images by means of a normalized cross-correlation function computed in the 2D spatial Fourier domain. They are also based on the principle that a relative shift in the spatial domain results in a linear phase term in the Fourier domain. Some works [Pla and Bober, 1997], [Lucceche et al., 1997] have arisen using the frequency domain to deal with motion models more complex than translations. Frequency domain-based method are very useful when there exist a clear dominant motion in the scene, which appear clearly as a single pick in the frequency domain (see Figure 2.2). But the accuracy of the estimation is lower when it does not exist a clear dominant motion, which is the normal case in real problems. Frequency domain-based techniques have been used to obtain the initial motion parameters (frequently the translational parameters) which are used as initial estimation in more complex and accurate iterative motion estimation algorithms.

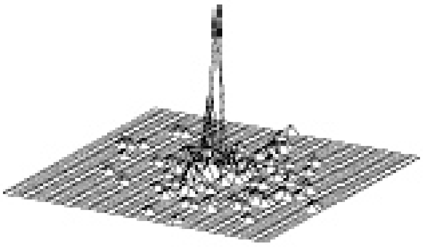


Figure 2.2: *An example of a peak in the discrete cross correlated function.*

2.3 Details on feature-based image registration

A feature-based image registration technique is used in our approach to obtain a good initial parameters that will be refined using the GLS-based motion estimation procedure. Due to the importance of the feature-based technique for achieving good results, we explain with a certain grade of detail this step of the proposed registration algorithm.

As it was be commented previously, the feature-based registration is usually carried out in three steps:

1. The first step consists of selection/extraction of features on the images.
2. Next, each feature in one image is compared with potential corresponding features in the other image. A pair of features with similar attributes are accepted as matches and are called control points.
3. Finally, the parameters of the best transformation which models the deformation between the images are estimated using the control points obtained in the previous step.

2.3.1 Step 1: Feature detectors

The input of this step are two images I_1 and I_2, and the output are two sets of feature points Ψ_1 and Ψ_2 obtained from the fist and second image respectively, which are defined as follows:

$$\begin{aligned}\Psi_1 &= \{[(x_1, y_1), s_1, \alpha_1], [(x_2, y_2), s_2, \alpha_2], \ldots, [(x_{n_1}, y_{n_1}), s_{n_1}, \alpha_{n_1}]\},\\ \Psi_2 &= \{[(x'_1, y'_1), s'_1, \alpha'_1], [(x'_2, y'_2), s'_2, \alpha'_2], \ldots, [(x'_{n_2}, y'_{n_2}), s'_{n_2}, \alpha'_{n_2}]\},\end{aligned} \tag{2.1}$$

where n_1 and n_2 are the number of feature points detected at each image and (x_i, y_i), s_i, α_i and (x'_j, y'_j), s'_j, α'_j are the localization, scale and orientation of the feature point at first and second images, respectively, $\forall i \in (1, \dots, n_1)$ and $\forall j \in (1, \dots, n_2)$.

A corner can be defined as the intersection of two edges. It can also be defined as a point for which there are two dominant and different edge directions in a local neighborhood of the point. An interest point (or feature point) is a point in an image which has a well-defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal. In practice, most of so-called corner detection methods detect interest points in general rather than corners in particular. As a consequence, if only corners are to be detected it is necessary to do a local analysis of detected interest points to determine which of these are real corners. Unfortunately, in the literature, "corner", "interest point" and "feature" are used somewhat exchangeable. We prefer to use the terminology of feature or interest point rather than corner, since it is more general.

In an ideal scenario, many feature points detected at the first image should have their corresponding position at second one, regardless the type and the degree of the deformation between both images. Therefore, the feature detector must be invariant to translations, scale changes, rotations, illuminations changes, viewpoint changes, etc. The techniques able to perform this work are called *Invariant Feature Detectors*. To make an extensive revision of the most popular invariant feature detector is not an objective of this work. There are two papers that describe and compare some of the most important techniques, see [Schmid et al., 2000] and [Mikolajczyk et al., 2005]. The second one is more recent and deals with affine invariant detectors.

The *SIFT* interest point detector

One of the most popular invariant feature detectors is the *Scale Invariant Feature transform* (*SIFT*) [Lowe, 2004] which has been included as a tool of the proposed image registration algorithm. The main idea of the *SIFT* technique, as well as other invariant feature detectors, can be showed in Figure 2.3. The same feature point can be detected in both images and, in addition, it is also possible to determine the size of the area of interest of the point. This size is related to the *scale* where the interest point has been detected. Both areas of interest represent the same information. Besides of *SIFT* detector is invariant to scale changes, it is also invariant to any degree of rotations, some illumination changes and even

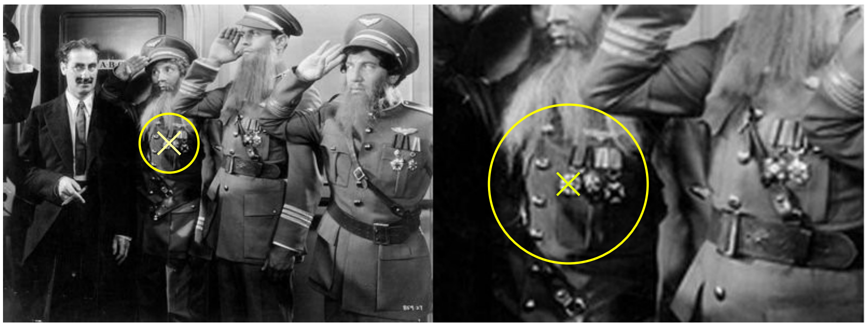


Figure 2.3: *The invariant feature detectors are able to detect the same interest point in images with scale changes. In addition, the size of the neighbor of interest is also estimated. The feature point in the left image has been detected at a different scale than the one of the right image (the zoomed). But both represent the same neighborhood of the point.*

moderate viewpoint changes.

The sift detector has three main steps:

1. **Scale-space extrema detection:** In this stage, the interest points are detected. For this purpose, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken. Interest points are then taken as maxima/minima of the difference of gaussians (DoG) that occur at multiple scales.

   Once DoG images have been obtained, interest points are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate interest point.

   This interest point detection step is a variation of one of the blob detection methods proposed by Lindeberg [Lindeberg, 1998] by detecting scale-space maxima of the scale normalized Laplacian, that is detecting points that are local extrema with respect to both space and scale, in the discrete case by comparisons with the nearest 26 neighbours in a discretized scale-space volume. The difference of Gaussians operator can be seen as an approximation to the Laplacian.

2. **Interest point localization:** Scale-space extrema detection produces too many interest points candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This information allows to reject points that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

3. **Orientation assignment:** In this step, each interest point is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation since the interest point descriptor can be represented relative to this orientation and therefore achieving invariance to image rotation.

   The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the interest point in the Gaussian-blurred image where it was detected. An orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the interest point. The peaks in this histogram correspond to dominant orientations. Once the histogram is filled, the orientation corresponding to the highest peak is assigned to the interest point. In addition, the orientation of any local peaks that are within 80% of the highest peaks is also assigned to the interest point. In the case of multiple orientations being assigned, an additional interest point is created having the same location and scale as the original interest point for each additional orientation.

Previous steps found interest point locations at particular scales and assigned orientations to them. This ensured invariance to image location, scale and rotation.

2.3.2 Step 2: Feature descriptors and matching

This process is applied to all the interest points detected at both images, obtaining two sets of descriptors Π_1 and Π_2, which are defined as follows:

$$\begin{aligned} \Pi_1 &= \{\pi_1, \pi_2, \ldots, \pi_{n_1}\} \\ \Pi_2 &= \{\pi'_1, \pi'_2, \ldots, \pi'_{n_2}\} \end{aligned} \tag{2.2}$$

where π_i and π'_j ($i \in (1, \ldots, n_1)$ and $j \in (1, \ldots, n_2)$), are the descriptors related to feature points $[(x_i, y_i), s_i, \alpha_i]$ and $[(x'_i, y'_i), s'_i, \alpha'_i]$, respectively.

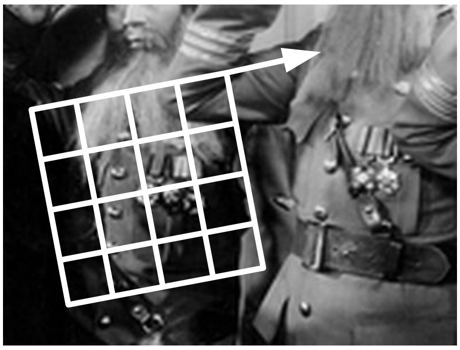


Figure 2.4: *A grid of* 4×4 *cells is centered at each feature point. The cell is oriented using the orientation of the feature point. The size of the cell depends on the scale of the feature point. At each cell, a histogram of orientations with* 8 *bins is calculated. The final descriptor has* 128 *dimensions.*

As well as with feature detectors, there are several works about feature description. See [Mikolajczyk and Schmid, 2005] for a comprehensive comparative. We focus on *SIFT* because is the technique that has been used in this work.

The *SIFT* technique not only detect interest points. It also describe the points in an invariant way such that the descriptors are highly distinctive and partially invariant to the remaining variations, like illumination, $3D$ viewpoint, etc.

To describe each previously detected interest point, a 4×4 grid is putted on and centered at the localization of the point. To allow rotation invariance, the grid is oriented using the orientation of the feature point α_i. The size of the grid depends obviously on the scale of the point s_i. The gray level of the pixels into the grid are normalized to be invariant to illumination changes. With the resulting values, the descriptor is calculated using a histogram of 8 orientations at each cell of the grid (see Figure 2.4). The resulting descriptor has 128 dimensions, since there are 4×4 cells and for each cell 8 values are obtained. This vector is normalized to improve invariance to changes in illumination.

Comparison of SIFT features with other local features

There has been an extensive study done on the performance evaluation of different local descriptors, including SIFT, using a range of detectors [Mikolajczyk and Schmid, 2005]. The main results are summarized as follows:

- SIFT and SIFT-like GLOH (Gradient Location and Orientation Histogram) features exhibit the highest matching accuracies for an affine transformation of 50 degrees rotations. After this transformation limit, results start getting unreliable.

- Distinctiveness of descriptors is measured by summing the eigenvalues of the descriptors, obtained by the Principal components analysis of the descriptors normalized by their variance. This corresponds to the amount of variance captured by different descriptors, therefore, to their distinctiveness. PCA-SIFT (Principal Components Analysis applied to SIFT descriptors), GLOH and SIFT features give the highest values.

- SIFT-based descriptors outperform other local descriptors on both textured and structured scenes, with the difference in performance larger on the textured scene.

- For scale changes in the range 2-2.5 and image rotations in the range 30 to 45 degrees, SIFT and SIFT-based descriptors again outperform other local descriptors with both textured and structured scene content.

- Performance for all local descriptors degraded on images introduced with a significant amount of blur, with the descriptors that are based on edges, like shape context, perform increasingly poorly with increasing amount of blur. This is because edges disappear in the case of a strong blur. But GLOH, PCA-SIFT and SIFT still performed better than the others. This is also true for evaluation in the case of illumination changes.

The evaluations carried out suggests strongly that SIFT-based descriptors, are the most robust and distinctive, and are therefore best suited for feature matching. That is the main reason for using the *SIFT* techniques in this work.

SIFT implementation

There are several implementation accessible in Internet, some of them are the followings:

- Original David Lowe's implementation (C/Matlab): `http://www.cs.ubc.ca/~lowe/keypoints/`

- Krystian Mikolajczyk's one used at paper [Mikolajczyk and Schmid, 2005] (C): `http://www.robots.ox.ac.uk/~vgg/research/affine/`

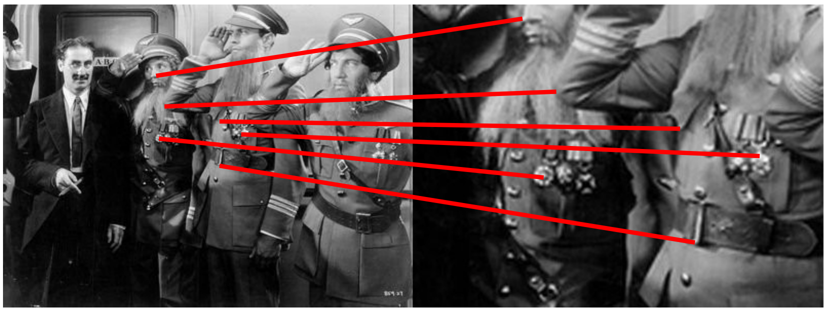


Figure 2.5: *An hypothetic example of feature points matched. Red lines represent that a descriptor located at one position of the first image matches with a descriptor located at the second image.*

- Sebastian Nowozin (C#): http://user.cs.tu-berlin.de/~nowozin/libsift/
- Andrea Vedaldi (Matlab/C): http://vision.ucla.edu/~vedaldi/code/sift/sift.html
- Andrea Vedaldi (C++): http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html

Matching descriptors

The next step is to find correspondences between descriptors of the first image with descriptors belonging to the second one. That its, to find a set of matches: $\{\pi_i \leftrightarrow \pi'_j\}$, $i \in (1, \ldots, n_1)$ and $j \in (1, \ldots, n_2)$. Figure 2.5 shows an hypothetic example of that, red lines represent that a descriptor located at one position of the first image matches with a descriptor located at the second image.

One of the most popular technique is to perform a Nearest Neighbour (NN) search strategy where for each descriptor π_i from the set Π_1, the most similar descriptor (based on a distance function) from the set Π_2 is looked for. As an alternative, a k-NN search strategy can be used where instead of looking for the most similar, the k-th most similar are obtained.

2.3.3 Step 3: Motion parameters estimation

From the previous step, a set of matches $\{\pi_i \leftrightarrow \pi'_j\}$, $i \in (1, \ldots, n_1)$ and $j \in (1, \ldots, n_2)$ has been obtained. The output of this step is the vector of motion parameters χ that best explain the transformation between images. For that purpose, a random sampling algorithm (like RANSAC, [Fischler and Bolles, 1981]) can be used since there is a high likelihood that, in spite of using invariant descriptors and sophisticated matching techniques, many feature points have been incorrectly matched. In fact, it is very probable that only a few number of matches are useful.

The random sampling techniques consist of randomly getting the minimum number of matches needed to estimate the motion parameters (3 for affine motion and 4 for projective) and to estimate the motion. This process is repeated many times. For each try, a cost function is calculated which measure the quality of the estimation. The best try is the one that has the best cost functions. The parameters associated to that try are selected as the estimated motion parameters.

In affine motion, with only 3 matches $\pi_1 \leftrightarrow \pi_1'$, $\pi_2 \leftrightarrow \pi_2'$ and $\pi_3 \leftrightarrow \pi_3'$, the motion parameters can be easily estimated a system of 6 equations as follows:

$$\left\{ \begin{array}{l} x_1' = a_1 x_1 + b_1 y_1 + c_1 \\ y_1' = a_2 x_1 + b_2 y_1 + c_2 \\ x_2' = a_1 x_2 + b_1 y_2 + c_1 \\ y_2' = a_2 x_2 + b_2 y_2 + c_2 \\ x_3' = a_1 x_3 + b_1 y_3 + c_1 \\ y_3' = a_2 x_3 + b_2 y_3 + c_2 \end{array} \right. \tag{2.3}$$

That equations can be written in $A\chi = b$ form as follows:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \end{pmatrix} \tag{2.4}$$

Then, the vector of motion parameters cab be estimated by $\chi = A^{-1}b$.

Random sampling algorithms

Random sampling and consensus (RANSAC) is one of the most widely used robust estimator in computer vision today. Since it was introduced at [Fischler and Bolles, 1981] some improvement have been proposed. In [Torr and Zisserman, 1997], it was pointed out that RANSAC treats all inliers uniformly. In other words, in the cost function, all the inliers score a null cost, while all outliers score a constant penalty. Better performance was obtained by using a cost function where the inliers scores a penalty depending on how well it satisfies the required functional relation while the outliers score a constant penalty. This new idea is known as MSAC (M-estimator sample consensus) [Torr and Zisserman, 1997] and it was found to give better performance than the original RANSAC without processing time penalization. A slightly different algorithm was proposed in [Torr and Zisserman, 2000], where the cost function was modified to yield the maximum likelihood estimate under the assumption that outliers are uniformly distributed. The algorithm was called MLESAC (maximum likelihood sampling and consensus).

In this work MSAC has been used, due to the next reasons:

- SIFT algorithm provides a lots of good interest points. RANSAC-based techniques are good enough to deal with our problem. Note that the RANSAC technique is only used to estimate a first approximation of the motion parameters. After that, an optimization technique will be used to refine the estimation.
- MLESAC gets better results than RANSAC but it takes more processing time to perform the task.
- We prefer MSAC since it gets better results than RANSAC but without taking more processing time to perform the task.

Cost functions

To measure the quality of estimation at each try, a cost function is used. At original RANSAC the cost function counts the number of inliers of each try. The best estimate is the one with the large number of inliers. In MSAC, the cost function penalize the inliers according to its distance to the model and the outliers using a constant value. In this case, the best estimate is the one with less cost value.

Lets suppose that a set of matches $\{\pi_i \leftrightarrow \pi'_j\}$, $i \in (1,\ldots,n_1)$ and $j \in (1,\ldots,n_2)$ has been obtained. The descriptors π_i belong to the first image and the descriptors π'_j belong to the second one. Let us also suppose that the pixel coordinates of a particular match are (x_i, y_i) and (x'_j, y'_j), χ is the estimated motion parameters in a try of the random sampling process and $\phi((x_i, y_i), \chi)$ is a function which apply the motion parameters to the input coordinates from the first image to obtain the coordinates of that point at the second image.

Then, if the parameters χ have been well estimates, the coordinates obtained with $\phi((x_i, y_i), \chi)$, i.e. $(\hat{x}_i, \hat{y}_i)$, have to be very close to the coordinates of the point

```
Input: Two Images $I_1$, $I_2$.
Output: $\chi$, the vector of estimated motion parameters.
Use SIFT detector to obtain the set of interest points $\Psi_1$ and $\Psi_2$.
For each interest point, use SIFT descriptor to obtain the set of descriptors $\Pi_1$ and $\Pi_2$.
Apply a knn search strategy to find the matches between the set of descriptors.
$j = 0$.
repeat
    Get randomly 3 matches for affine motion (4 for projective).
    Solve the equation system $A\chi_j = b$ (see Equation 2.4) using the selected matches.
    Obtain cost $d_j$ evaluating the cost function of the estimated $\chi_j$.
    $j = j + 1$.
until $j < N$
$\chi = \chi_k$, where $k$ is the best try, i.e. $d_k = \min(d_j), \forall j = 1 \dots N$.
```

Figure 2.6: *Feature-based (SIFT-based) image registration algorithm*

at second image. Therefore, the registration error function $reg_{error}((x_i, y_i), (x'_j, y'_j), \chi)$ can be calculated as follows:

$$reg_{error}((\hat{x}_i, \hat{y}_i), (x'_j, y'_j), \chi) = \sqrt{(\hat{x}_i - x'_j)^2 + (\hat{y}_i - y'_j)^2} \tag{2.5}$$

2.3.4 Feature-based image registration algorithm

In the previous sections an algorithm to perform image registration by using a feature-based technique (SIFT-based) has been explained in detail. The steps can be summarized at the algorithm showed in Figure 2.6, where N is the number of tries of the random sampling algorithm used.

Figures 2.7 and 2.8 show a real example of performance of this algorithm. Figure 2.7 shows the results of the SIFT procedure where small green squares are the localization of the detected feature points. In addition, the size of the area of interest of the point has been drawn using a yellow circle. As can be seen, a lot of feature points are detected in both images. Figure 2.8 shows the results of the matching procedure. There are many feature points that have been correctly

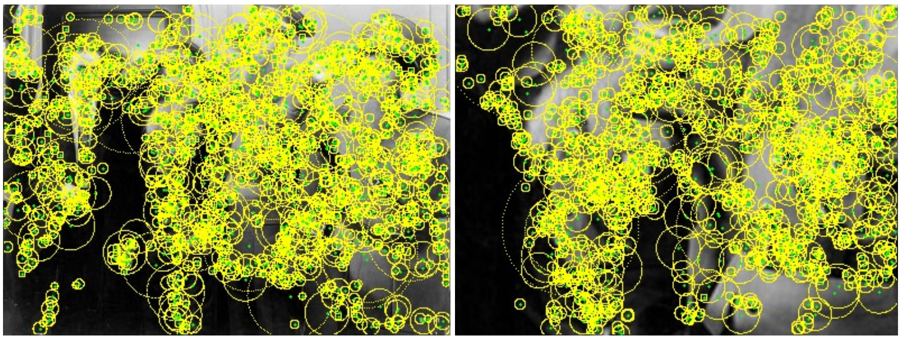


Figure 2.7: *Real example of SIFT performance. The position of each detected featured point is showed using a small green square. The area of influence of each feature point is drawn using a yellow circle. As this example shows, many interest points are obtained at each image.*

matched, but there are also some other matches that are not correct. It has been commented before that in spite of the excellent behavior of the SIFT technique, not all the matches are correct. The MSAC algorithm deals with this situation obtaining a good estimate of the initial motion parameters.

2.4 GLS-based motion estimation

We assume that the input of the motion estimation problem are two images I_1 and I_2, and the output is the vector of motion parameters χ that best explains the transformation between both images. First, the GLS method is briefly explained for general fitting problems. Subsequently, the GLS algorithm for motion estimation is described in detail.

2.4.1 GLS for general problems

In regression problems, we mainly deal with two types of residuals: residuals of the observations and residuals of the functions (see Figure 2.9). That yields two different definitions of the objective function Θ to be minimized: Θ_υ, based on the residuals of the observations, and Θ_ϵ, based on the residuals of the functions. In statistics terminology, the data regression under Θ_υ is referred to as geometric fitting, while Θ_ϵ as algebraic fitting. The minimization of Θ_υ provides a fitted model for which the sum of squares of the distances to the given observations is minimal. Hence, the residuals to the coordinate measurements are perpendicular

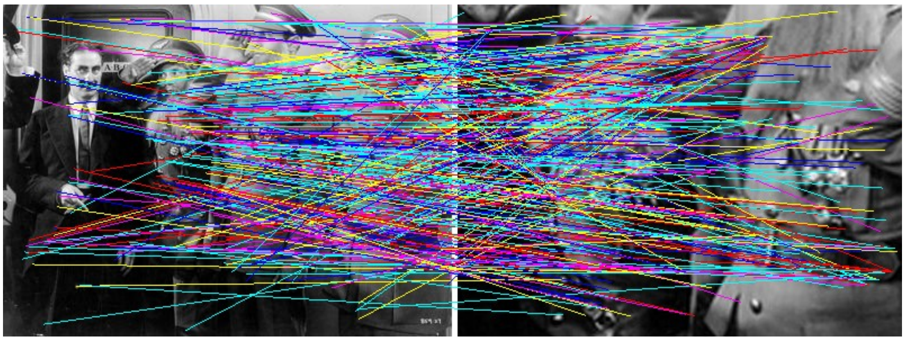


Figure 2.8: *Real example of the performance of the matching procedure. Many feature points have been correctly matched, but some other have been badly matched.*

to the fitted model, that is the reason why geometric fitting is also termed orthogonal distance regression (see Figure 2.9b). It is well known that the minimization of Θ_υ obtains better performance than Θ_ϵ in terms of accuracy, for instance see [Danuser and Stricker, 1998], [Zhang, 1997] or [Bad-Hadiashar and Suter, 1998]. The GLS estimator uses a Θ_υ objective function. Therefore, it can be considered that the GLS tries to solve the regression problem using orthogonal distances.

In general, the GLS estimation problem can be expressed as follows:

$$\text{minimize } [\Theta_\upsilon = \upsilon^T \upsilon] \text{ subject to } F(\chi, \lambda) = 0, \tag{2.6}$$

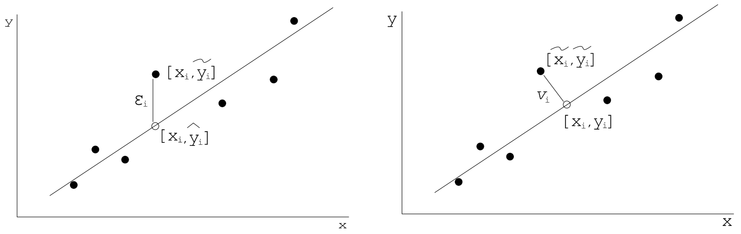
where:

- υ is a vector of r unknown residuals in the observation space, that is, $\upsilon = \lambda - \tilde{\lambda}$, where λ and $\tilde{\lambda}$ are the unperturbed and actually measured vector of observations, respectively.

- $\chi = (\chi^1, \dots, \chi^p)^T$ is a vector of p parameters;

- λ is made up by r elements λ_i, $\lambda = (\lambda_1, \dots, \lambda_r)^T$, each one is an observation vector with n components $\lambda_i = (\lambda_i^1, \dots, \lambda_i^n)^T$

- $F(\chi, \lambda)$ is made up by r elements $F_i(\chi, \lambda_i)$, $F(\chi, \lambda) = (F_1(\chi, \lambda_1), \dots, F_r(\chi, \lambda_r))^T$, each one is, in general, a set of f functions that depend on the common vector of parameters χ and on an observation vector λ_i, $F_i(\chi, \lambda_i) = (F_i^1(\chi, \lambda_i), \dots, F_i^f(\chi, \lambda_i))^T$. Those functions can be non-linear.



(a) Residuals of functions (b) Residuals of observations

Figure 2.9: *Example of the two types of residuals for a well known line fitting regression problem ($y = mx + n$). The black points are the input data and the solid line shows a possible estimation of the fitted line. The left image shows the concept of residual of functions where ϵ_i is the distance between the observed point $[x_i, \tilde{y}_i]$ and the estimated point $[x_i, \hat{y}_i]$ by the model. The right image shows the concept of residual of observations where υ_i is the distance between the observed point $[\tilde{x}_i, \tilde{y}_i]$ and the unperturbed point $[x_i, y_i]$.*

Note that the minimization problem has two unknowns, the parameters χ and the unperturbed observations λ.

Thus, the solution of (2.6) can be addressed as an iterative optimization starting with an initial guess of the parameters $\widehat{\chi}(0)$ and with a first estimate of the observation $\widehat{\lambda}(0) = \tilde{\lambda}$. At each iteration j, the algorithm estimates $\widehat{\Delta\chi}(j)$ to update the parameters as follows: $\widehat{\chi}(j) = \widehat{\chi}(j-1) + \widehat{\Delta\chi}(j)$. The process is stopped if the improvement $\widehat{\Delta\chi}(j)$ at iteration j is smaller than an user-specified resolution in the parameter space. Together with the improvement of the parameters, the estimates for the unperturbed observations are updated after each iteration step by $\widehat{\lambda}(j) = \widehat{\lambda}(j-1) + \widehat{\Delta\lambda}(j)$.

The minimization problem expressed in Equation 2.6 can be solved [Britt and Luecke, 1973] using the Lagrange formalism, which relies on the objective function:

$$\Phi = \upsilon^T \upsilon - 2k^T F(\chi, \lambda), \tag{2.7}$$

where k represents the vector of r Lagrange multipliers. To find the improvements $\Delta\chi(j)$ and $\Delta\lambda(j)$, Equation 2.7 is linearized around the current estimates $\widehat{\chi}(j)$ and $\widehat{\lambda}(j)$ resulting in

$$\Phi \approx \upsilon^T \upsilon - 2k^T (A\Delta\chi + B\Delta\lambda - E), \tag{2.8}$$

where $A = \partial F/\partial \chi$, $B = \partial F/\partial \lambda$ and $E = -F(\widehat{\chi}(j), \widehat{\lambda}(j))$. The estimates $\widehat{\Delta\chi}(j)$ and $\widehat{\Delta\lambda}(j)$ are then obtained by solving the equation system

$$[\partial\Phi/\partial\Delta\lambda, \partial\Phi/\partial\Delta\chi, \partial\Phi/\partial k] = 0. \tag{2.9}$$

The elimination of k in Equation 2.9 yields the desired expressions of $\widehat{\Delta\chi}(j)$ and $\widehat{\Delta\lambda}(j)$

$$\widehat{\Delta\chi}(j) = (A^T Q A)^{-1} A^T Q E, \tag{2.10}$$

$$\widehat{\Delta\lambda}(j) = B^T Q (I - A(A^T Q A)^{-1} A^T Q) E, \tag{2.11}$$

where the matrix $Q = (BB^T)^{-1}$ has been introduced to simplify the notation. Equations 2.10 and 2.11 can also be expressed in a more convenient way as follows:

$$\widehat{\Delta\chi}(j) = \left(\sum_{i=1 \ldots r} N_i \right)^{-1} \left(\sum_{i=1 \ldots r} R_i, \right), \tag{2.12}$$

$$\widehat{\Delta\lambda_i}(j) = B_i^T (B_i B_i^T)^{-1} (E_i - A_i \widehat{\Delta\chi}(j)), \forall i, \tag{2.13}$$

where $N_i = A_i^t (B_i B_i^t)^{-1} A_i$ and $R_i = A_i^t (B_i B_i^t)^{-1} E_i$, with

$$B_i = \begin{pmatrix} \frac{\partial F_i^1(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \lambda_i^1} & \ldots & \frac{\partial F_i^1(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \lambda_i^n} \\ \vdots & & \vdots \\ \frac{\partial F_i^f(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \lambda_i^1} & \ldots & \frac{\partial F_i^f(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \lambda_i^n} \end{pmatrix}_{(f \times n)}, \tag{2.14}$$

$$A_i = \begin{pmatrix} \frac{\partial F_i^1(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \chi^1} & \ldots & \frac{\partial F_i^1(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \chi^p} \\ \vdots & & \vdots \\ \frac{\partial F_i^f(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \chi^1} & \ldots & \frac{\partial F_i^f(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1))}{\partial \chi^p} \end{pmatrix}_{(f \times p)}, \tag{2.15}$$

$$E_i = \begin{pmatrix} -F_i^1(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1)) \\ \vdots \\ -F_i^f(\widehat{\chi}(j-1), \widehat{\lambda}_i(j-1)) \end{pmatrix}_{(f \times 1)}. \tag{2.16}$$

2.4.2 An example of using the GLS for plane estimation

In this section, a plane estimation example is explained to best understand the GLS estimation procedure. The input of the problem is a set of $3D$ points $(x_i, y_i, z_i), i = 1 \ldots N$, where N is the number of points. The aim is to estimate the best plane that fits with the input data points. The plane equation is expressed as follows:

$$F(\chi, \lambda_i) = ax_i + by_i + cz_i + d = 0, \forall i \tag{2.17}$$

where $[x_i, y_i, z_i]$ are the observation data (i.e. $\lambda_i = (x_i, y_i, z_i)$) and $[a, b, c, d]$ are the unknown parameters (i.e. $\chi = (a, b, c, d)^T$).

In order to apply the GLS estimation procedure, the matrices B_i, A_i and E_i must to be calculated. For this problem, the matrices can be expressed as follows:

$$\begin{aligned}
B_i &= \left(\frac{\partial F_i(\chi, \lambda_i)}{\partial x_i}, \frac{\partial F_i(\chi, \lambda_i)}{\partial y_i}, \frac{\partial F_i(\chi, \lambda_i)}{\partial z_i} \right) = (a, b, c)_{(1 \times 3)} \\
A_i &= \left(\frac{\partial F_i(\chi, \lambda_i)}{\partial a}, \frac{\partial F_i(\chi, \lambda_i)}{\partial b}, \frac{\partial F_i(\chi, \lambda_i)}{\partial c}, \frac{\partial F_i(\chi, \lambda_i)}{\partial d} \right) = (x_i, y_i, z_i, 1.0)_{(1 \times 4)} \\
E_i &= -F_i(\chi, \lambda_i) = -(ax_i + by_i + cz_i + d)_{(1 \times 1)}
\end{aligned} \tag{2.18}$$

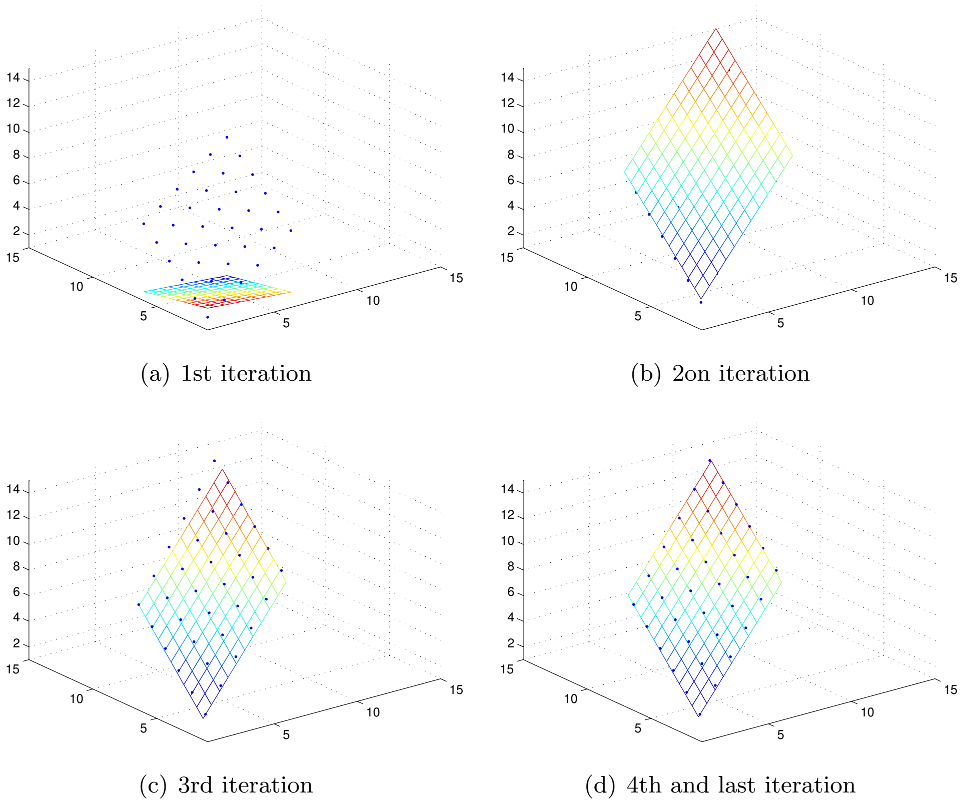
Figure 2.10 shows an example of the GLS estimation procedure. From top to bottom, left to right, this figure illustrates how the estimated plane has been progressively adjusted to the input data (blue points).

As it was be explained before, the estimating process starts with an initial guess of the parameters $\widehat{\chi}(0)$ and with a first estimate of the observations $\widehat{\lambda}(0) = \tilde{\lambda}$. Then, at each iteration j, $\widehat{\Delta\chi}(j)$ and $\widehat{\Delta\lambda}(j)$ are estimated by Equations 2.12 and 2.13, respectively. Note that GLS process adjusts the estimation parameters and also the observations values.

2.4.3 A GLS-based model in motion estimation problems

Let us to introduce now the proposed model based on the GLS technique and applied to motion estimation problems.

In motion estimation problems, the objective function is usually based on the assumption that the gray level of all the pixels of a region $\Re$ remains constant between two consecutive images in a sequence, i.e. the Brightness Constancy Assumption (BCA).



(a) 1st iteration

(b) 2on iteration

(c) 3rd iteration

(d) 4th and last iteration

Figure 2.10: *GLS-based plane estimation example. From top to bottom, left to right, this Figure illustrates how the estimated plane has been progressively adjusted to the input data (blue points). The estimated error is also progressively reduced as follows: a)* 183.06*, b)* 13.32 *c)* 0.27 *and finally d)* 0.15*.*

In order to directly use the BCA instead of its linearized version, i.e. the optical flow equation, a non-linear estimator should be used. The GLS estimator can be applied in this context. In our formulation of the motion estimation problem, the function $F_i(\chi, \lambda_i)$ is expressed as follows (note that in this case the number of functions f is 1):

$$F_i(\chi, \lambda_i) = I_1(x_i, y_i) - I_2(x'_i, y'_i), \tag{2.19}$$

where $I_1(x_i, y_i)$ is the gray level of the first image in the sequence (test image) at the point (x_i, y_i), and $I_2(x'_i, y'_i)$ is the gray level of the second image in the sequence (reference image) at the transformed point (x'_i, y'_i). In this case, each observation vector λ_i is related to each pixel (x_i, y_i), with r being the number of pixels in the area of interest.

Let us consider the reference image (I_2) as the data model to match, and the test image (I_1) as observation data. For each pixel i, let us define the observation vector as:

$$\lambda_i = (x_i, y_i, I_1(x_i, y_i)), \tag{2.20}$$

which has three elements $(n = 3)$: column, row (pixel coordinates) and gray level of reference image at these coordinates. The gray level of the test image has been selected as an element of the observation vector since it is the observed gray level that we want to match with some gray level in the reference image using the BCA. The spatial coordinates have also been selected as part of the observations, since inaccuracy in their measurement can happen, because of the image acquisition process.

As explained further on, this observation model will lead to obtain a constraint in the optimization process expressed as a set of weights, which will measure the influence of each observation in the estimation process using image gradient information.

In order to calculate the matrices A_i, B_i and E_i (see Equations 2.14, 2.15, and 2.16), the partial derivatives of the function $F_i(\chi, \lambda_i)$ with respect to the parameters and with respect to the observations must be worked out. The partials of the functions $F_i(\chi, \lambda_i)$ with respect to the parameters χ^m, $(m = 1 \dots p)$ are calculated using the chain rule and can be expressed as follows:

$$\begin{aligned} \frac{\partial F_i(\chi, \lambda_i)}{\partial \chi^m} &= \frac{\partial I_1(x_i, y_i)}{\partial \chi^m} - \frac{\partial I_2(x'_i, y'_i)}{\partial \chi^m} = 0 - \frac{\partial I_2(x'_i, y'_i)}{\partial \chi^m} \\ &= -\left(\frac{\partial I_2(x'_i, y'_i)}{\partial x'_i} \frac{\partial x'_i}{\partial \chi^m} + \frac{\partial I_2(x'_i, y'_i)}{\partial y'_i} \frac{\partial y'_i}{\partial \chi^m} \right) \\ &= -\left(I_2^x(x'_i, y'_i) \frac{\partial x'_i}{\partial \chi^m} + I_2^y(x'_i, y'_i) \frac{\partial y'_i}{\partial \chi^m} \right), \end{aligned} \tag{2.21}$$

where $I_2^x(x'_i, y'_i)$ and $I_2^y(x'_i, y'_i)$, are the components of the gradient of the reference image at the pixel (x'_i, y'_i) in x and y direction. The expressions $\frac{\partial x'_i}{\partial \chi^m}$ and $\frac{\partial y'_i}{\partial \chi^m}$ will be calculated using a specific motion model.

On the other hand, the partials of the functions $F_i(\chi, \lambda_i)$ with respect to a particular element λ_i^l, $(l = 1 \dots n)$ of the observation vector λ_i is calculated using the chain rule and can be expressed as follows:

$$\begin{aligned}
\frac{\partial F_i(\chi,\lambda_i)}{\partial \lambda_i^l} &= \frac{\partial I_1(x_i,y_i)}{\partial \lambda_i^l} - \frac{\partial I_2(x_i',y_i')}{\partial \lambda_i^l} \\
&= \frac{\partial I_1(x_i,y_i)}{\partial \lambda_i^l} - \left(\frac{\partial I_2(x_i',y_i')}{\partial x_i'}\frac{\partial x'}{\partial \lambda_i^l} + \frac{\partial I_2(x_i',y_i')}{\partial y_i'}\frac{\partial y'}{\partial \lambda_i^l}\right) \\
&= \frac{\partial I_1(x_i,y_i)}{\partial \lambda_i^l} - \left(I_2^x(x_i',y_i')\frac{\partial x_i'}{\partial \lambda_i^l} + I_2^y(x_i',y_i')\frac{\partial y_i'}{\partial \lambda_i^l}\right).
\end{aligned} \tag{2.22}$$

Analogously, the expressions $\frac{\partial x_i'}{\partial \lambda_i^l}$ and $\frac{\partial y_i'}{\partial \lambda_i^l}$ will be calculated using a specific motion model.

The vector of parameters χ depends on the motion model used. For affine and projective motion, the vector of parameters are $\chi = (a_1,b_1,c_1,a_2,b_2,c_2)^T$, $(p=6)$ and $\chi = (a_1,b_1,c_1,a_2,b_2,c_2,d,e)^T$, $(p=8)$ respectively. The transformed coordinates (x_i',y_i') are related to the original ones (x_i,y_i) in affine (Equation 2.23) and projective (Equation 2.24) motion as follows:

$$\left\{\begin{array}{l} x_i' = a_1x_i + b_1y_i + c_1 \\ y_i' = a_2x_i + b_2y_i + c_2 \end{array}\right. \tag{2.23}$$

$$\left\{\begin{array}{l} x_i' = \dfrac{a_1x_i + b_1y_i + c_1}{dx_i + ey_i + 1} \\ y_i' = \dfrac{a_2x_i + b_2y_i + c_2}{dx_i + ey_i + 1} \end{array}\right. \tag{2.24}$$

Therefore, the terms B_i, A_i and E_i are expressed for affine (Equation 2.25) and projective (Equation 2.26) motion as follows:

$$\begin{aligned}
B_i &= (I_1^x - a_1I_2^x - a_2I_2^y, I_1^y - b_1I_2^x - b_2I_2^y, 1.0)_{(1\times3)} \\
A_i &= (-x_iI_2^x, -y_iI_2^x, -I_2^x, -x_iI_2^y, -y_iI_2^y, -I_2^y)_{(1\times6)} \\
E_i &= -\big(I_1(x_i,y_i) - I_2(x_i',y_i')\big)_{(1\times1)}
\end{aligned} \tag{2.25}$$

$$\begin{aligned}
B_i &= (I_1^x - I_2^xN_1 - I_2^yN_2, I_1^y - I_2^xN_3 - I_2^yN_4, 1.0)_{(1\times3)} \\
A_i &= \frac{-1}{N_d}(x_iI_2^x, y_iI_2^x, I_2^x, x_iI_2^y, y_iI_2^y, I_2^y, N_5, N_6)_{(1\times8)} \\
E_i &= -\big(I_1(x_i,y_i) - I_2(x_i',y_i')\big)_{(1\times1)}
\end{aligned} \tag{2.26}$$

where I_1^x, I_1^y, I_2^x and I_2^y have been introduced to simplify notation as:

$$\begin{aligned} I_1^x &= I_1^x(x_i, y_i) \\ I_1^y &= I_1^y(x_i, y_i) \\ I_2^x &= I_2^x(x_i', y_i') \\ I_2^y &= I_2^y(x_i', y_i') \end{aligned} \tag{2.27}$$

being $I_1^x(x_i, y_i)$, $I_1^y(x_i, y_i)$, the components of the gradient of the test image at point (x_i, y_i); and $I_2^x(x_i', y_i')$ and $I_2^y(x_i', y_i')$ the components of the gradient of the reference image at point (x_i', y_i'). In addition, N_d, N_1, N_2, N_3, N_4, N_5 and N_6 have also been introduced as follows:

$$\begin{aligned} N_d &= (dx_i + ey_i + 1) \\ N_1 &= \frac{a_1 N_d - d(a_1 x_i + b_1 y_i + c_1)}{N_d^2} \\ N_2 &= \frac{a_2 N_d - d(a_2 x_i + b_2 y_i + c_2)}{N_d^2} \\ N_3 &= \frac{b_1 N_d - e(a_1 x_i + b_1 y_i + c_1)}{N_d^2} \\ N_4 &= \frac{b_2 N_d - e(a_2 x_i + b_2 y_i + c_2)}{N_d^2} \\ N_5 &= x_i x_i' I_2^x + x_i y_i' I_2^x \\ N_6 &= y_i x_i' I_2^x + y_i y_i' I_2^x \end{aligned} \tag{2.28}$$

The complete GLS motion estimation algorithm is summarized in the Figure 2.11. Note that both the parameters and observations are updated at each iteration. The inclusion of the estimation of $\widehat{\Delta\lambda_i}(j)$ for each observation introduces a significant computational complexity to the algorithm, since the gradients of the images should be estimated at each iteration. This is not an easy work, since the neighborhood relations among the pixels are hard to estimate when pixel coordinates have been moved from an iteration to another.

An alternative to the algorithm showed in Figure 2.11 is to fix the observation values and not to estimate at each iteration the increment of the observations. By doing that, the search space of solutions is restricted, since the objective function 2.6 is optimized allowing to vary it in the χ space, instead of the $(\chi,\ \lambda)$ space. However, very satisfactory solutions can be obtained in an acceptable computing time, although maybe not as good as the one provided by the complete algorithm.

```
Input: Images $I_1$, $I_2$ and the initial motion parameters $\chi(0)$
Output: $\widehat{\chi}$, the vector of estimated motion parameters.
Calculate image gradients: $I_1^x$, $I_1^y$, $I_2^x$, $I_2^y$.
$j = 0$.
Set up observation vectors $\lambda_i(0), \forall i$ using the current measured
  values $(x_i, y_i, I_1(x_i, y_i))$.
repeat
    $j = j + 1$.
    Update matrices $A_i$, $B_i$ and $E_i$ using $\widehat{\chi}(j-1)$ and $\lambda_i(j-1), \forall i$
    Estimate $\widehat{\Delta\chi}(j)$.
    Estimate $\widehat{\Delta\lambda_i}(j)$ for each observation.
    $\widehat{\chi}(j) = \widehat{\chi}(j-1) + \widehat{\Delta\chi}(j)$.
    $\widehat{\lambda}_i(j) = \widehat{\lambda}_i(j-1) + \widehat{\Delta\lambda_i}(j)$ for each observation.
    Reestimate image gradients $I_1^x$, $I_1^y$, $I_2^x$, $I_2^y$.
until $|\widehat{\Delta\chi}(j)|$ is small enough.
$\widehat{\chi} = \widehat{\chi}(j)$
```

Figure 2.11: *Complete generalized least squares motion estimation algorithm*

The simplified motion estimation procedure is summarized in a new algorithm showed in Figure 2.12, which is the algorithm used in the experiments reported in this thesis.

2.4.4 The role of the weights in GLS-based motion estimation

It has been commented before that the increment of the motion parameters are estimated using the GLS method by the next Equation:

$$\widehat{\Delta\chi}(j) = (A^T(BB^T)^{-1}A)^{-1}A^T(BB^T)^{-1}E, \tag{2.29}$$

In the IRLS technique (see [Odobez and Bouthemy, 1995] for more details) the increment of the parameters is estimated by the following expression:

$$\widehat{\Delta\chi}(j) = (J^TWJ)^{-1}J^TWd, \tag{2.30}$$

where J is the jacobian of the objective function with respect to the motion parameters, d is the vector of independent terms and W is a diagonal matrix, which is used as weight matrix where each w_i measures the influence of the i^{th} observation in the global estimation of the parameters. The IRLS method

```
**Input:** Images $I_1$, $I_2$ and the initial motion parameters $\chi(0)$
**Output:** $\widehat{\chi}$, the vector of estimated motion parameters.
Calculate image gradients: $I_1^x$, $I_1^y$, $I_2^x$, $I_2^y$.
$j = 0$.
Set up observation vectors $\lambda_i(0), \forall i$ using the current measured
values $(x_i, y_i, I_1(x_i, y_i))$.
**repeat**
    $j = j + 1$.
    Update matrices $A_i$, $B_i$ and $E_i$ using $\widehat{\chi}(j-1)$ and $\lambda_i(j-1), \forall i$
    Estimate $\widehat{\Delta\chi}(j)$.
    $\widehat{\chi}(j) = \widehat{\chi}(j-1) + \widehat{\Delta\chi}(j)$.
    $\widehat{\lambda}_i(j) = \widehat{\lambda}_i(j-1)$ for each observation.
**until** $|\widehat{\Delta\chi}(j)|$ is small enough.
$\widehat{\chi} = \widehat{\chi}(j)$
```

Figure 2.12: *Simplified generalized least squares motion estimation algorithm*

starts assigning to all w_i the same value for all observations, i.e. $w_i = 1, \forall i$. Therefore, at the first iteration, all the observations have the same influence in the estimation. After the parameters have been updated, the weights are also updated based on the residuals of the objective function E_i by:

$$w_i = \psi_i(E_i)/E_i, \tag{2.31}$$

where ψ is the influence function of the M-Estimator used in the method.

For instance, when using the well-known Tukey M-Estimator [Tukey, 1977], the weights for each observation are updated as follows:

$$w_i = \begin{cases} \frac{(C^2 - E_i^2)^2}{E_i} \text{ if } |E_i| < C \\ 0 \text{ otherwise} \end{cases} \tag{2.32}$$

where C is a tuning constant.

Note that in the proposed formulation (Equation 2.29), analogously to IRLS method (Equation 2.30), the expression $(B_i B_i^T)^{-1}$ plays the role of a weight providing high values (close to 1.0) when the gradient values in the reference and the test image are similar, and low ones (close to 0.0) in the opposite case. For instance, if we consider a translating motion model (i.e. an affine motion model with $a_1 = b_2 = 1.0$ and $a_2 = b_1 = 0.0$), the weights are expressed as follows:

$$w_i = (B_i B_i^T)^{-1} = \frac{1}{(I_1^x - I_2^x)^2 + (I_1^y - I_2^y)^2 + 1}, \tag{2.33}$$

where I_1^x, I_1^y, I_2^x and I_2^y have been introduced to simplify notation as has been previously expressed in Equation 2.27.

Note that when the motion parameters are correctly estimated, the values of the gradients for a given pixel in the reference and in the test image will have very similar values, thus w_i will be close to 1.0. In the opposite case, with pixels having different gradient values in the test and in the reference image, w_i will be close to 0.0, reducing their influence in the estimation.

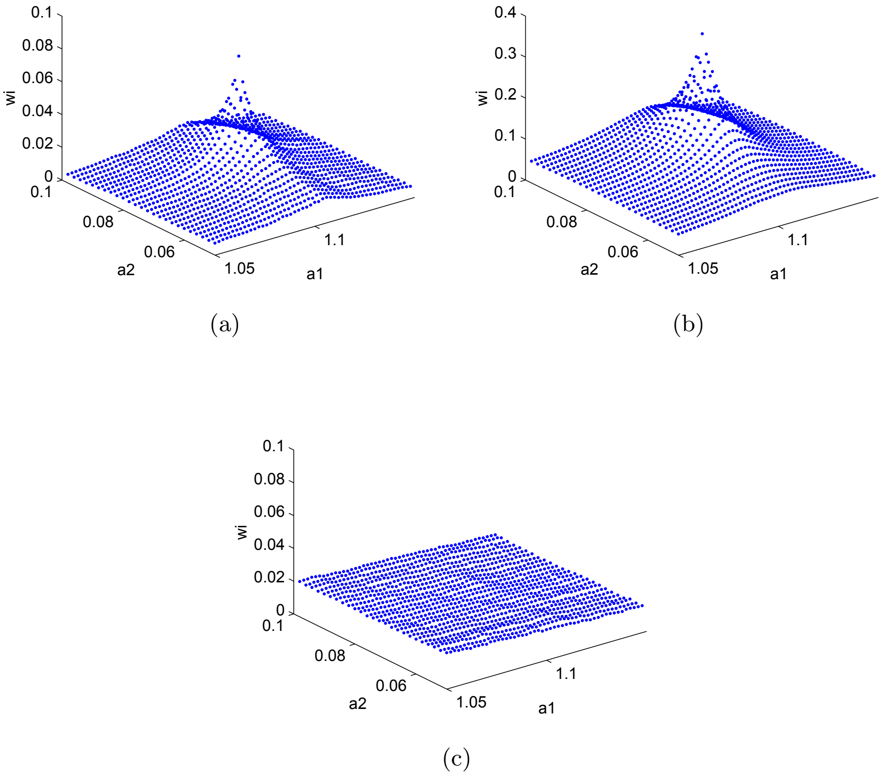
Similar considerations can be deduced when the complete affine motion model is used. In this case the weights are expressed as follows:

$$w_i = (B_i B_i^T)^{-1} = \frac{1}{(a_1 I_1^x + a_2 I_1^y - I_2^x)^2 + (b_1 I_1^x + b_2 I_1^y - I_2^y)^2 + 1} \tag{2.34}$$

One of the main differences between both methods is how the weights are updated at each iteration. While the weights of the IRLS techniques are based on the residuals of the objective function (i.e. based on gray levels differences), the weights at the proposed method are based on gradient differences, which presents a more invariant behavior against some important source of deformations. For instance, in the presence of intensity illumination changes. It is well-known that gradient information is a key factor in motion estimation. In fact, the areas with more information for motion estimation are the ones that are located at the object edges of the image.

In order to illustrate the behavior of the weights, an experiment with a synthetic sequence has been created with two different movements. The left part of the image has undergone an affine transformation (in particular, $a_1 = 1.108$ and $a_2 = 0.08$, see equation 2.23) and the right part remains static. The observations have been splitted into three sets:

- **Set** A: Observations belonging to the left part of the image where gradient values are significant.
- **Set** B: The rest of the observations of the left part, that is, where the gradient values are low, which could represent low textured areas.
- **Set** C: Observations belonging to the right part, which have a different motion from sets A and B.



(a)

(b)

(c)

Figure 2.13: *Average of weights for: (a) set A, (b) set B, (c) set C (see text). For the sets A and B, the proposed method produces high weights when the parameters estimated are close to the true ones. In this example, the true parameters are $a_1 = 1.108$, $a_2 = 0.08$. Note that set C does not follow this behavior.*

The weights w_i of all the observations of the three sets have been calculated for a range of possible values of the affine parameters a_1 and a_2 around the true values. Figures 2.13(a,b,c) show the average of the weights obtained for each combination of the affine parameters a_1 and a_2 in the considered range, for the sets A, B and C, respectively. In the case of sets A and B, i.e. the pixels belonging to the left part of the image, notice that when the parameters a_1 and a_2 are close to the real ones, the average of the weights are higher than when the parameters are far from the correct value. Indeed, the maximum weight value is reached exactly at the correct motion values.

Note that the maximum of the set A is smaller than the one of the set B due to the fact that the gradients in the set A have larger magnitude than the ones in the set B, but in both cases the magnitude of weights when the parameters are close to the real ones is significantly bigger than in the case when the parameters are far from the correct ones. That is, when the parameters are close to a valid solution, the weight of a pixel in the left part of the image reaches a local maximum.

However, the observations belonging to the set C (the right part of the image) do not follow this behavior, having always low weights values, since none of the combinations of the affine parameters a_1, a_2 considered, are correct for the motion corresponding to this part of the image. A similar behavior has been detected when using a projective motion model.

2.5 Two-steps image registration algorithm

In many image registration problems where the deformation between images is quite large (e.g. large rotation, very different viewpoints, strong changes of scale, etc.), it is necessary to initialize the motion estimator using a good initial motion parameters. In the proposed Image Registration algorithm, first a feature-based method is used to obtain a good initial motion parameters that are not very far from the true solution, in order to try to avoid falling in a local minima. Using this initialization, in the second step, the GLS-based motion estimator is applied, which refines the estimation of the motion parameters up to the accuracy level desired by the user.

At the first step, to cope with strong changes of scale, any rotation degree, illumination changes and partially affine invariance, a SIFT-based technique [Lowe, 2004] has been used to detect and describe interest points. Thus, the proposed image registration algorithm can be summarized in these two sequential steps:

1. **Feature-based initialization**: The Section 2.3 explains in detail this step. The algorithm showed in the Figure 2.6 summarizes the tasks to be performed in this step.

2. **Final motion estimation using GLS**: The proposed GLS-based motion estimator is applied using as observations all the pixels into the overlapped area of both images. The algorithm showed in Figure 2.12 is used in this step.

2.6 Experiments and results

2.6.1 Experiments using the affine motion model

The first experiment tests the accuracy of the proposed GLS-based motion estimation algorithm and compares it with two IRLS-based motion estimators. The first one is the well-known Odobez and Bouthemy's motion estimator, which is a good representative IRLS technique that uses M-Estimators to deal with outliers (see [Odobez and Bouthemy, 1995] for details). This motion estimator will be called *Motion2D* hereafter. The second one is the Baker et. al.'s Inverse Compositional Iteratively Reweighted Least Squares motion estimator which is a modification of the Inverse Compositional motion estimator for achieving robustness. See [Baker and Maththews, 2004], [Baker and Matthews, 2002], [S. Baker and Ishikawa, 2003] for details. This motion estimator will be called *RIC* (Robust Inverse Compositional) hereafter.

The Inverse Compositional algorithm has the main advantage that it is more computationally efficient than other motion estimation algorithms. However, its robust version looses this advantage since the weights have to be re-estimated at each iteration. In [S. Baker and Ishikawa, 2003], two more efficient robust algorithms were also presented, but both are an approximation to the *RIC* estimator and therefore they can not obtain so accurate estimates as the original *RIC*. The *RIC* algorithm has been used at this comparative experiments, since we care about the accuracy of the estimates.

Among the different approaches in motion estimation for image registration, the algorithms based on robust M-estimators (and in general based on adding a weight at each observation) are the most representative and successful in the literature, from the point of view of achieving accurate estimations in presence of noise and outliers in real image registration applications. Therefore, we use these motion estimator as one of the references for comparison purposes. In addition, the authors' implementations have been used [1] for a fair comparison.

For comparison purposes, all motion estimation techniques must be tested in equal conditions, therefore the featured-based step is performed first using the original input images I_1 and I_2 (i.e. the first step of the Image Registration algorithm reported at Section 2.5). Then, using the motion parameters obtained, the second image is transformed. Thus, the resulting image, I_{aux} should not be very far from I_1. At this stage, all motion estimation techniques are applied to

[1] Odobez and Bouthemy's algorithm source code is available at `http://www.irisa.fr/vista/Motion2D/`. Baker et. al.'s algorithms source code is available at `http://www.ri.cmu.edu/projects/project_515.html`

estimate the deformation between images I_1 and I_{aux}.

On the one hand, the proposed GLS-based estimator and the *RIC* one are initialized using the null motion vector, that is $x'_i = x_i$ and $y'_i = y_i$, $\forall i$ (i.e. $a_1 = b_2 = 1.0$ and $b_1 = c_1 = a_2 = c_2 = 0.0$ for affine motion). On the other hand, the *Motion2D* estimator uses its own hierarchical technique to avoid falling in a local minimum, in addition to the already corrected motion by the feature-based initialization. Somehow, this fact would give the *Motion2D* estimator a certain degree of advantage over the proposed GLS-based and the *RIC* one, which do not use a multiresolution approach.

Error measures

To check the accuracy of the registration, five error measures have been calculated using the pixels of the overlapped area of both images. They are the Normalized Correlation Coefficient (NCC), the Mean Absolute predictor Error (MAE), the Increment Sign Correlation coefficient (ISC [Kaneko et al., 2002]), the selective correlation Coefficient (SCC [Kaneko et al., 2003]) and finally the Structural SIMilarity index ($SSIM$ [Wang et al., 2004]).

The absolute value of NCC lies between 0 (low similarity) and 1 (high similarity). The NCC is expressed as follows, with μ_1,μ_{aux} being the average of the gray level of both images and $\Re$ the overlapped area:

$$NCC(I_1, I_{aux}) = \frac{\sum_{(x_i,y_i)\in\Re}[(I_1 - \mu_1)(I_{aux} - \mu_{aux})]}{\sqrt{\sum_{(x_i,y_i)\in\Re}(I_1 - \mu_1)^2 \sum_{(x_i,y_i)\in\Re}(I_{aux} - \mu_{aux})^2}}. \tag{2.35}$$

I_1 and I_{aux} have been introduced to simplify notation as: $I_1 = I_1(x_i, y_i), I_{aux} = I_{aux}(x'_i, y'_i)$.

The MAE provides values from 0 (high similarity) to ∞ (low similarity). It is defined as follows:

$$MAE(I_1, I_{aux}) = \text{mean}_i|I_1 - I_{aux}|. \tag{2.36}$$

The NCC is preferred over MAE because of its invariance to linear brightness and constant variations. Both measures perform badly in the presence of non-linear pixel brightness variations due to illuminations variations, occlusions and shadows. In order to obtain better performance Kaneko et. al. ([Kaneko et al., 2002], [Kaneko et al., 2003]) developed two new measures, the ISC and the SCC. Both measures lie between 0 (low similarity) and 1 (high similarity). The ISC algorithm is based on the average evaluation of incremental tendency of brightness in adjacent pixels. It first converts the list of pixel brightness values to a list of corresponding binary codes $\{b_1^I, b_2^I, \ldots, b_{n-1}^I\}$. They are defined for an image I as follows:

$$b_i^I = \begin{cases} 1 \text{ if } I(i+1) \geq I(i) \\ 0 \text{ otherwise} \end{cases} \tag{2.37}$$

The ISC between binary codes $b_i^{I_1}$, $b_i^{I_{aux}}$ obtained from images I_1 and I_{aux} is defined as follows:

$$ISC(I_1, I_{aux}) = \frac{1}{n}\sum_{i=1}^{n}(b_i^{I_1} b_i^{I_{aux}} + (1-b_i^{I_1})(1-b_i^{I_{aux}})) \tag{2.38}$$

The SCC method is an extension of the NCC method with a masking function for corresponding pixels in both images. Thus, only some selected pixels contribute to similarity computation. It is defined as follows:

$$SCC(I_1, I_{aux}) = \frac{\sum_{(x_i,y_i)\in\Re} c_i[(I_1-\mu_1)(I_{aux}-\mu_{aux})]}{\sqrt{\sum_{(x_i,y_i)\in\Re} c_i(I_1-\mu_1)^2 \sum_{(x_i,y_i)\in\Re} c_i(I_{aux}-\mu_{aux})^2}}. \tag{2.39}$$

The mask coefficient c_i represents the similarity of sign increment in the adjacent pixels in both images. It is defined using the binaries codes $b_i^{I_1}$, $b_i^{I_{aux}}$ as follows:

$$c_i = \begin{cases} 1 - |b_i^{I_1} - b_i^{I_{aux}}| \text{ if } i = 0 \text{ or even} \\ c_{i-1} \text{ otherwise} \end{cases} \tag{2.40}$$

The $SSIM$ index can be viewed as a quality measure of one of the images being compared (I_{aux}), using the other image as its version with perfect quality (I_1). Under the assumption that human visual perception is highly adapted for extracting structural information from a scene, Wang et. al. [Wang et al., 2004] introduced a framework for quality assessment based on the degradation of structural information. The use of this measure has shown clear advantages over traditional error measures like MAE and NCC in some specific problems. The reader is referenced to [Wang et al., 2004] for a comprehensive study of this technique. The $SSIM$ index also lies between 0 (low similarity) and 1 (high similarity). A Matlab implementation of this index is available online [2]

[2] http://www.ece.uwaterloo.ca/~z70wang/research/ssim/

Image gradient estimation

The estimation of the gradient of the images is one of the main points for achieving accuracy. To estimate the gradients, a Gaussian derivative operator has been applied. Then, to obtain sub-pixel values at image gradients, bilinear interpolation has been used.

Test images

A set of challenging sets of image pairs have been selected. They can be downloaded from Oxford's Visual Geometry Group web page [3]. They present four types of changes between images in 5 different sets of images: Blur: *Bikes* set, global illumination: *Leuven* set, jpg compression: *Ubc* set and zoom+rotation: *Bark* and *Boat* sets.

The scale change (*Bark* and *Boat* sets) and blur (*bikes* set) sequences were acquired by varying the camera zoom and focus respectively. The scale changes by about a factor of four. The light changes (*leuven* set) are introduced by varying the camera aperture. The JPEG sequence (*Ubc* set) is generated using a standard *xv* image browser with the image quality parameter varying from 40% to 2%. Each image set has 6 different images. A sample of the images is showed in Figure 2.14. the complete image sets are showed in Appendix A. The affine motion model has been used in all the sets.

Note that these sets of images have been captured in real conditions with the exception of the *Ubc* set where the different jpg compressions have been created by a synthetic way. For instance, there are people that appear in some images of *Boat* set and their position have changed or even disappeared in other ones. In the *Leuven* and *Bikes* sets there is also a small spatial variation due to the fact that the images were captured at different times. In the *bark* example, not all the images have the same illumination conditions because of the fact that the light is reflected on the bark of the tree from a different angle in some images of the set.

Results

For each set, the 6 images have been combined in all possible pairs ($1-2$, $1-3$, $\ldots$, $1-6$, $2-3$, $\ldots$, $5-6$). Figure 2.15 shows the accuracy results obtained using the three methods. Although the three methods obtain good results, the proposed GLS-based motion estimation obtains more accurate results than the other two methods in practically all cases for all error measures. In the *Ubc* and

[3] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

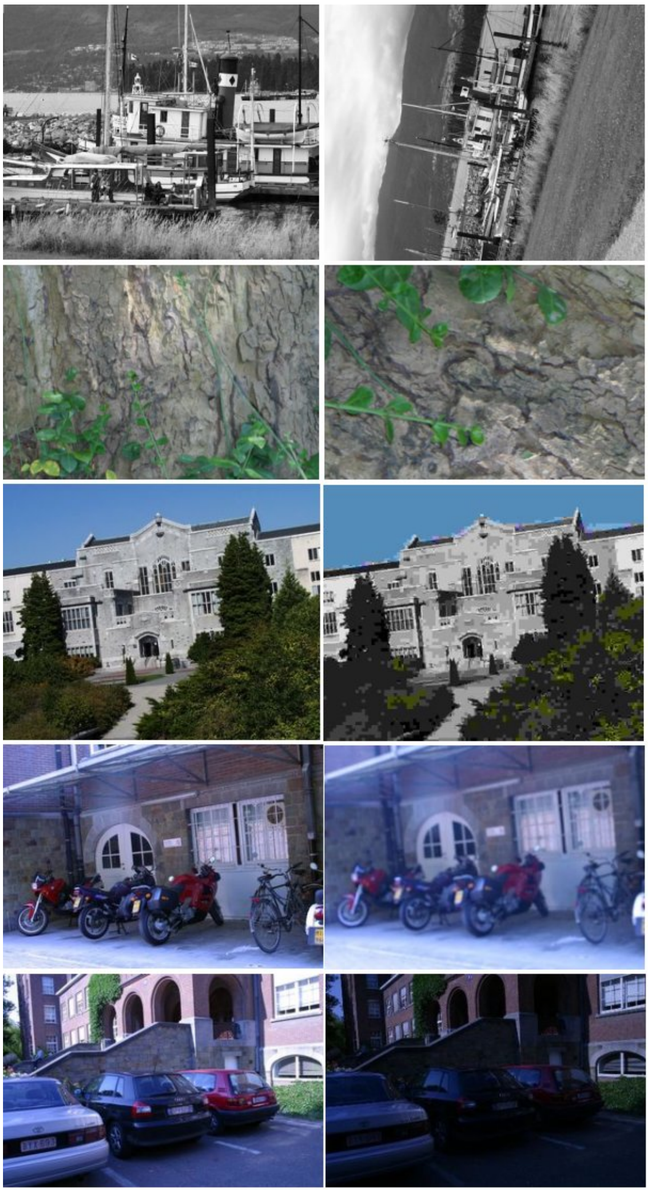


Figure 2.14: *Sample images (up to bottom) from* Boat, Bark, Ubc, Bikes *and* Leuven *image sets.*

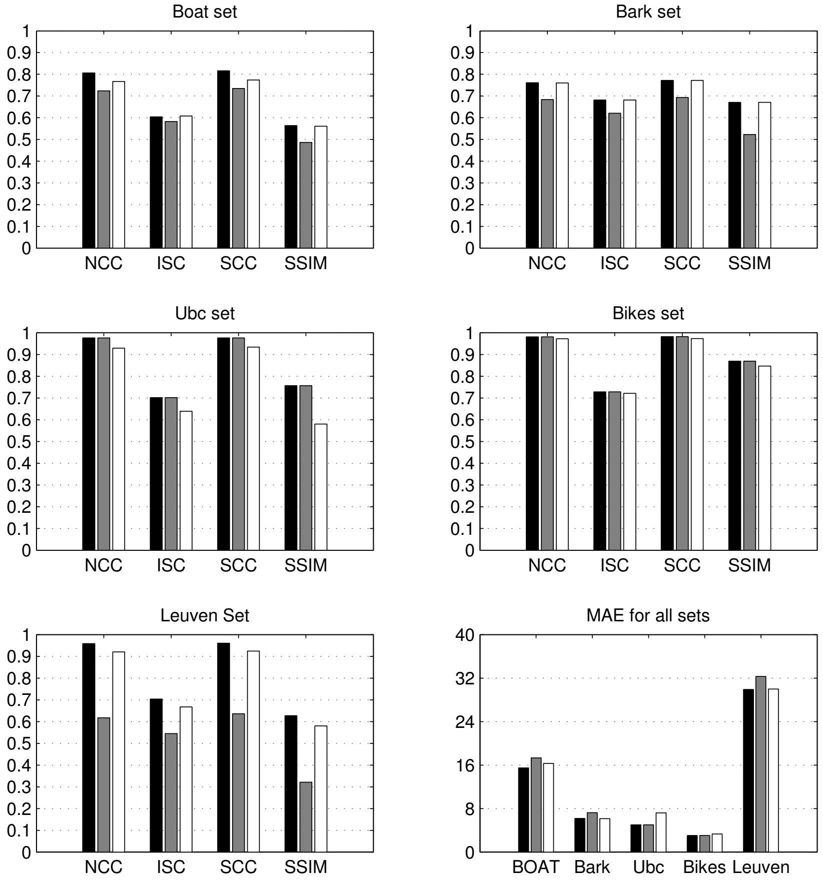


Figure 2.15: *From top to down, left to right: NCC, ISC, SCC and SSIM estimated using the three methods methods (proposed GLS-based black bar,* Motion2D *grey bar, RIC white bar) for* Boat, Bark, Ubc, Bikes *and* Leuven *image set. The last plot shows the MAE estimated using the three methods methods for all image sets.*

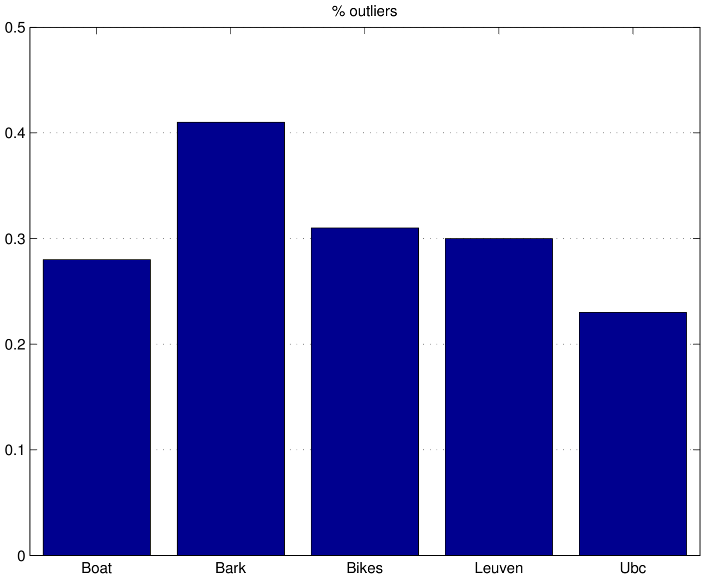


Figure 2.16: *Percentage of outliers introduced at each set.*

Bikes sets the three algorithms practically obtain the same results due to the deformation between images of these sets have not a significative magnitude and therefore the three methods can easily deal with these kind of deformation.

Figure 2.17 shows the results obtained for both algorithms when additional outliers are introduced to the images. In particular, an image patch of 200×200 have been introduced to image I_{aux}. Figure 2.16 shows the percentage of artificial outliers introduced to each image set. The results obtained show that the three algorithms can deal with that important number of outliers. The accuracy level of all techniques decreases with the presence of outliers but they still maintain a high accuracy level. The proposed GLS-based estimator continues obtaining more accurate results in most cases, showing in this case that, despite the fact that the proposed GLS formulation can not be considered as a robust estimator in statistical terms, it can obtain even better estimates than using robust estimators like the M-Estimators used in *Motion2D* and *RIC* algorithms. A collection of image registration results are showed in Appendix B. In particular, Figures B.2 and B.2 show results of *Boat* image set; Figures B.3 and B.4 show results of *Bark* image set; and Figures B.5, B.6 and B.7 show results of *Bikes*, *Ubc*, and *Leuven* image sets, respectively.

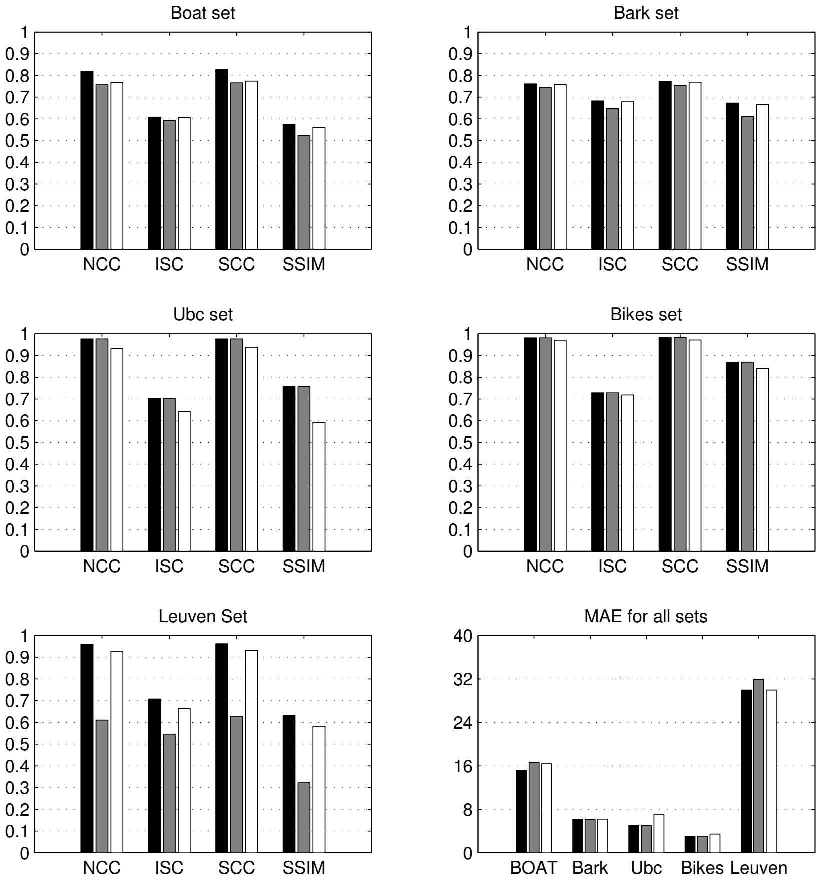


Figure 2.17: *Accuracy results when an image patch of* 200 × 200 *has been introduced as a source of outliers. Proposed GLS-based black bar,* Motion2D *grey bar and RIC white bar.*

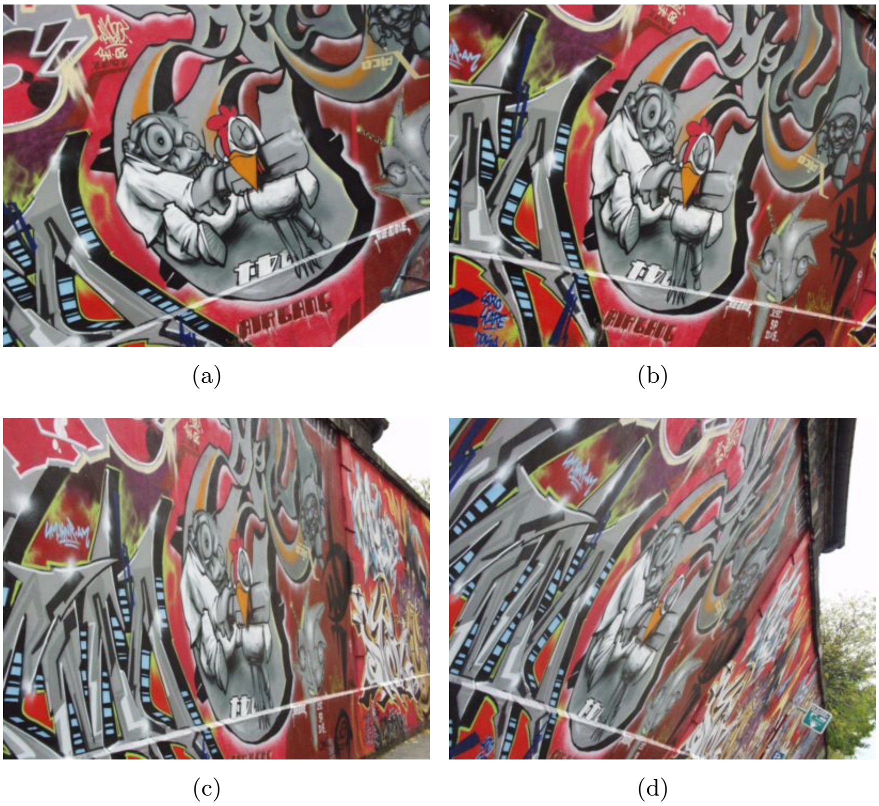
2.6.2 Experiments using the projective motion model

To test our approach using the projective motion model the *Graf* challenging image set has been used (It can also be download from Oxford's Visual Geometry Group web page). Sample images from this image set are showed in Figure 2.18 and the complete image set is showed in Figure A.6. The main deformation is due to a strong viewpoint changes. In fact, the camera varies from a fronto-parallel view to one with significant foreshortening at approximately 60 degrees to the camera. In addition, this image set has an additional difficulty. There is a white car placed in some images (See the bottom-right corner at Figure 2.18a), in other images its position changed, and even it disappeared in some other images. Thus, the pixels belonging to the car are a new source of outliers.

Unfortunately, the original *Motion2D* and *RIC* software provided by authors do not implement the projective motion model (needed for *Graf* image set). Therefore, it is not possible to perform a comparison with these methods using that motion model. We present the results obtained for *Graf* set using the proposed motion estimator for the projective motion model.

The proposed image registration algorithm has been applied to images from the *Graf* set obtaining accurate estimation of the motion parameters. Some illustrative registration results are showed in Figures 2.19 and 2.20 as image mosaics. Additional registration results are showed in Appendix B. To obtain these images, the image registration algorithm reported in Section 2.5 (i.e. feature-based + GLS-based) has been used. The image I_2 has been transformed using the motion parameters estimated. This image has been combined with the first image to create the panoramic image averaging the corresponding pixel values. The accuracy of the registration can be more appreciated at image edges (see Figures 2.19 and 2.20). Note how the pixels belonging to the car do not disturb the accurate estimation of the motion parameters thanks to the weights that arise from the proposed formulation of the motion estimation problem. The observations related with those pixels have been considered as outliers (i.e. its weight values are very close to 0) during the GLS-based process and therefore they have not influenced the estimation of the real motion parameters.

In order to test the proposed approach with other type of images, an experiment using real satellite images has been performed. Figure 2.21 shows two input satellite images from the same area but they have been captured at different times/days and therefore with different illumination conditions. Figure 2.22 shows the registration results as image mosaic. Just like the previous experiments, the registration has been performed obtaining accurate estimates. Some other image registration results are showed in Appendix B.



(a)

(b)

(c)

(d)

Figure 2.18: *Sample images from* graf *set. The changes between images are mainly due to the presence of a strong viewpoint change.*

2.7 conclusions

In this chapter, a new Generalized least squares-based motion estimator has been introduced. The proposed formulation of the motion estimation problem provides an additional constraint that helps to match the pixels using image gradient information. That is achieved thanks to the use of a weight for each observation, providing high weight values to the observations considered as inliers, i.e. the ones that support the motion model, and low values to the ones considered as outliers. The main characteristics of the proposed method are summarized as follows:

- It uses a non-linear GLS-motion estimation technique. Therefore, the BCA can directly be used instead of its linearized version, the optical flow equa-

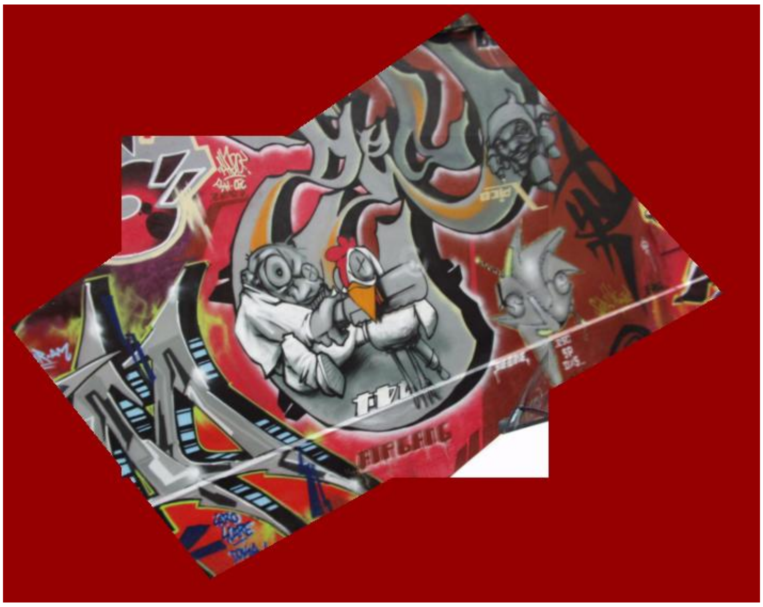


Figure 2.19: *Registration results for images from* graf *set.*

tion.

- To avoid falling in a local minimum, it uses a Feature-based method (SIFT-based) to obtain good initial motion parameters. Therefore it can deal with large motions.
- The GLS-based motion estimation technique includes an additional constraint, using gradient information as a way to deal with outliers.
- Similarly to the IRLS technique, the constraint is expressed as a weight to each observation, that varies during the iterative process.

The accuracy of our approach has been tested using challenging real images using both affine and projective motion models. Two Motion Estimator techniques, which use M-Estimators to deal with outliers into a iteratively reweighted least squared-based strategy, have been selected to compare the accuracy of our approach. The results obtained have showed that the proposed motion estimator can obtain as accurate results as M-Estimator-based techniques and even better in most cases.

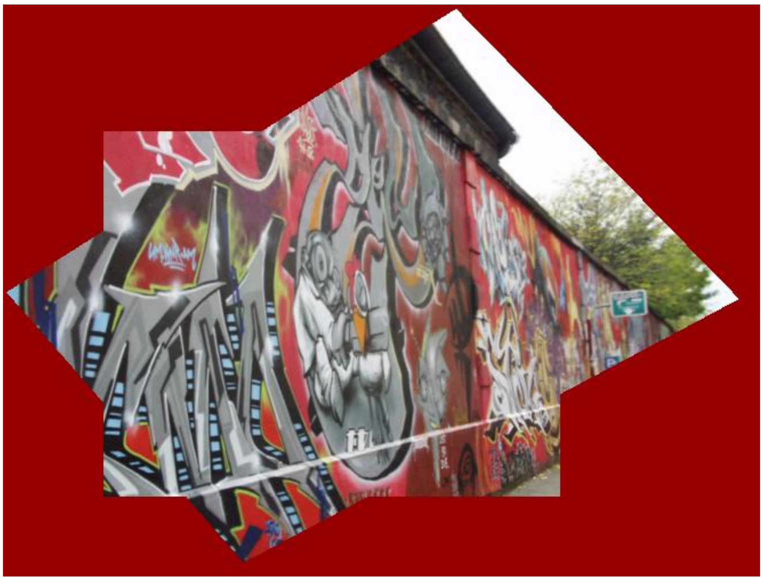
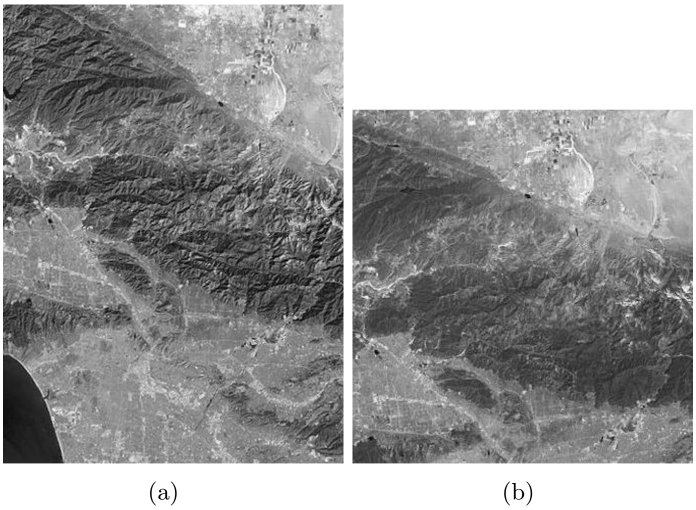


Figure 2.20: *Registration results for images from* graf *set.*



(a)

(b)

Figure 2.21: *Input satellite images.*

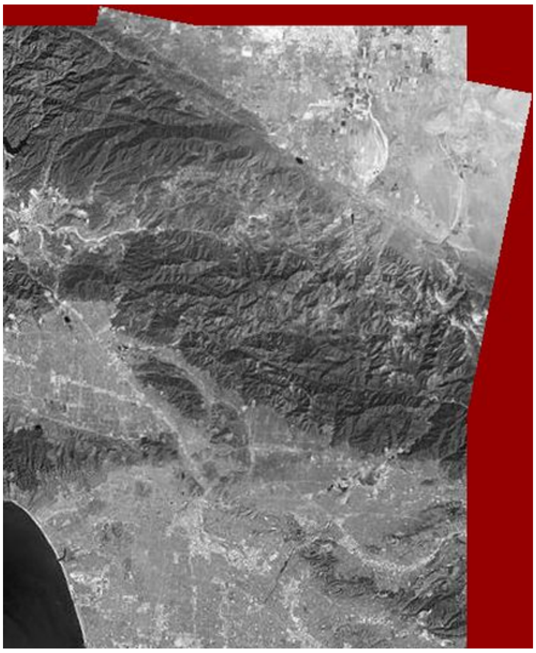


Figure 2.22: *Satellite image registration results.*

Chapter 3

Global motion estimation under non-uniform illumination changes

Contents

3.1 Introduction 66

3.2 Invariant color representations 67

3.3 Spatially varying illumination model 72

3.4 GLS-based global motion estimation under varying illumination 73

3.5 Experimental results 75

3.6 Conclusions 80

As it was pointed out in the previous chapter, the estimation of parametric global motion has had a significant attention during the last two decades, but despite the great efforts invested, there are still open issues. One of the most important ones is related to the ability to recover large deformation between images in the presence of illumination changes while keeping accurate estimates. In this chapter, the generalized least squared-based global motion estimator presented in Chapter 2 will be used in combination with a dynamic image model where the illumination factors are functions of the localization (x, y) instead of constants, allowing for a more general and accurate image model. Experiments using challenging images have been performed showing that the combination of both techniques is feasible and provides accurate estimates of the motion parameters even in the presence of strong illumination changes between the images.

3.1 Introduction

The estimation of motion in images is a basic task in computer vision with many application fields. One of the most important goals in the motion estimation field is to estimate the motion as accurately as possible. The problem of global motion estimation is not an easy task when there are large deformations and illumination variations between images. In addition, the presence of areas which do not support the main motion (outliers) is an additional source of inaccuracy.

Traditionally, the motion estimation problem has been formulated following the assumption that the changes in gray levels between images are only due to motion, i.e. the Brightness Constancy Assumption (*BCA*). A good example of a motion estimator based on the use of the *BCA* is the algorithm proposed in Chapter 2. However, a pixel can change its brightness value because an object moves to another part of the scene with different illumination or because the illumination of the scene changes, locally or globally, between images. In these cases, the *BCA* fails, and therefore, it is not possible to obtain accurate estimates.

It is important to make a difference between global and local illumination changes. On the one hand, global (or uniform) illumination changes refers to illumination changes that affect as the same manner to all the pixels of the image. For instance, a global illumination change can be produced if the aperture of the camera changes between images. The *Leuven* image set (see Figure A.5) is a good example of illumination changes produced by different apertures of the camera. On the other hand, local (or non-uniform) illumination changes refers to illumination changes that affect only to a part of the image or that can affect to all the pixels of the image but not in the same manner. For instance, the shadows of an object or the effects produced by a gradient-based illumination pattern.

To overcome the problem of estimating the motion in the presence of illumination changes, two are the most commonly used techniques. One technique consist of preprocessing the images to transform them to a new color space where shadows, highlights and other illumination effects have been partially removed [Finlayson et al., 2002], [Geusebroek et al., 2001], [Montoliu et al., 2005]. Then, the motion estimator is applied to the transformed images. Alternatively, a more complex image model than the *BCA* can directly be used in the motion estimation process. Thus, the estimator can calculate, at the same time, the motion and illumination parameters [Kim et al., 2004], [Lai and Fang, 1999]. The second type of approach is the technique that has been used in this work. In particular, a dynamic image model where the multiplication and bias illumination factors are functions of the localization (x, y) instead of constants, has been used at this work.

This dynamic image model has been combined with the Generalized Least Squares-based (*GLS*) global motion estimator presented at Chapter 2, which obtains accurate estimates even when there exist large deformations between images and in the presence of an important number of outliers. Therefore, with the combination of both techniques (the GLS-based global motion estimator and the use of the dynamic image model) a motion estimator can be obtained which can perform the motion estimation task in an accurate manner while allowing large deformation and non-uniform illumination changes between images. Thus, the main objective of this chapter is to reformulate the proposed global motion estimator that use a constant illumination model to accommodate it to non-uniform illumination changes by using a spatial dynamic image model.

In order to show a preview of the results obtained with the proposed approach and also to help to understanding the problem, the Figure 3.1 shows an illustrative example of how a non-uniform illumination change can affect to the accuracy of the estimation of the global motion when the *BCA* is used. The first row shows the two input images. They are the first and the fourth images from *Boat* image set (see Figure A.1). An artificial illumination change has been added to the right image. The second row shows a detail of the results of the registration procedure. The left one when using the original image registration algorithm (i.e. the one present in Chapter 2) and the right one when a dynamic image model is used instead of the *BCA*. Note how in the first case, the registration accuracy is quite low (for instance, see the funnel of the boat) while in the second case the accuracy level is much better.

The rest of the chapter is organized as follows: next section deals with color invariant representations, Section 3.3 explains the dynamic image model used in this paper, Section 3.4 comments how the dynamic image model has been combined with the GLS-based global motion estimator. Section 3.5 shows the main results and finally in the last section, the main conclusions drawn from this work are summarized.

3.2 Invariant color representations

The value of pixels belonging to a particular area of the image can vary when the light conditions change between images. For instance, viewing direction, surface orientation, highlights, illumination direction, illumination intensity and illumination color are some of the factors that can change the value of the pixels. The aim of invariant representations is to obtain the same value of a pixel independently of some of the previous factors. Figure 3.2 shows an illustrative example of how the invariant representation could help in image processing related prob-

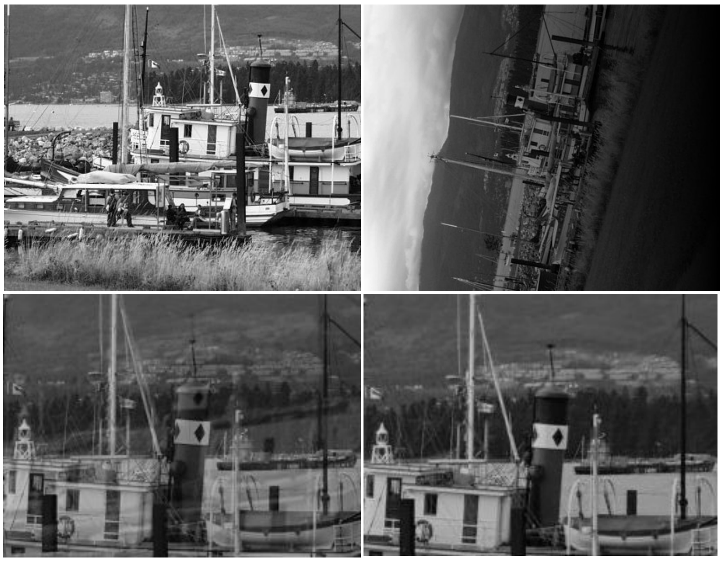


Figure 3.1: *This Figure shows an illustrative example of how a non-uniform illumination changes can affect to the accuracy of the estimation of the global motion when the* BCA *is used. The first row shows the two input images. An artificial illumination change has been added to the right image. The second row shows a detail of the results of the registration procedure. The left one when using the original image registration algorithm and the right one when a dynamic image model is used instead of the* BCA.

lems. The first row of Figure 3.2 shows a green toy and the results of applying a clustering technique using 2 classes and the RGB color as distance. In this image, the illumination conditions together with the position of the light source and the object produce shadows. The shadows of the toy are so dark that its distance to the background is less than its distance to the not shadowed pixels of the toy. Therefore, the shadowed part of the toy has been bad classified producing poor segmentation results. The second row shows the hypothetic results of applying a invariant function that eliminate the shadows and the results of the clustering process. After applying the invariant, all the pixels of the toy have approximatively the same color and therefore the clustering produces good segmentation results.

Invariant representations can be obtained by performing simple mathematical operations with the R, G and B bands [Gevers and Smeulders, 1999] such as band division and subtraction. The starting point is the Shafer's reflection model [Shafer, 1984] which explains how the pixel values depend on a set of factors.

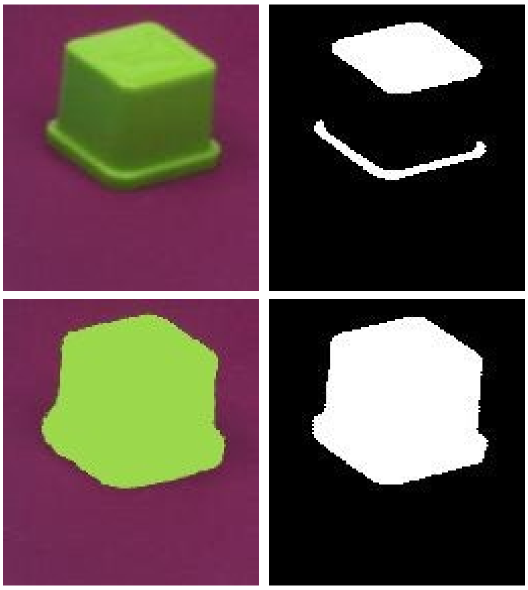


Figure 3.2: *First row: original image (with shadows) and the results of applying a clustering technique using only 2 clusters. Second row: results of applying the invariant representation that eliminates the shadows and the results of the same clustering process.*

The Shafer's reflectation model is expressed as follows:

$$C = m_b(\vec{n}, \vec{s}) \int_{\lambda} f_C(\lambda) c_b(\lambda) e(\lambda) d\lambda + m_s(\vec{n}, \vec{s}, \vec{v}) \int_{\lambda} f_C(\lambda) e(\lambda) c_s(\lambda) d\lambda \qquad (3.1)$$

for $C = \{R, G, B\}$ giving the Cth sensor response. Further, $f_R(\lambda)$, $f_G(\lambda)$ and $f_B(\lambda)$ are spectral sensitivities of the red, green and blue sensors, respectively, $c_b(\lambda)$ and $c_s(\lambda)$ are the surface albedo and Fresnel reflectance, respectively. λ denotes the wavelength, $e(\lambda)$ is the incident light, $\vec{n}$ is the surface patch normal, $\vec{s}$ is the direction of the illumination source and $\vec{v}$ is the direction of the viewer. Geometry terms m_b and m_s denote the geometric dependencies on the body and surface reflection components, respectively.

Considering the neutral interface reflection (NIR) model (assuming that $c_s(\lambda)$ has a constant value independent of the wavelength) and white illumination (equal energy density for all wavelengths whitin the visible spectrum), then $c_s(\lambda) = c_s$ and $e(\lambda) = e$ and hence being constants. Then, the Shafer's reflection model can be expressed as follows:

$$C = m_b(\vec{n},\vec{s})eK_C(\lambda) + m_s(\vec{n},\vec{s},\vec{v})ec_s f \tag{3.2}$$

where $K_C(\lambda)$ is the compact formulation depending on the sensors and the surface albedo only, and can be expressed as follows:

$$K_C(\lambda) = \int_\lambda f_C(\lambda)c_b(\lambda)d\lambda. \tag{3.3}$$

Further, if the integrated white condition holds, then:

$$f = \int_\lambda f_R(\lambda)d\lambda = \int_\lambda f_G(\lambda)d\lambda = \int_\lambda f_B(\lambda)d\lambda \tag{3.4}$$

For mate objects (i.e. when $m_s(\vec{n},\vec{s},\vec{v})ec_s f = 0$), the process of dividing two bands, i and j, produces a new representation that is invariant to object geometry and illumination intensity factors (see Equation 3.5), since $m_b(\vec{n},\vec{s})$ and e factors do not depend on the band:

$$\frac{C_i}{C_j} = \frac{m_b(\vec{n},\vec{s})eK_i(\lambda)}{m_b(\vec{n},\vec{s})eK_j(\lambda)} = \cancelto{1}{\frac{m_b(\vec{n},\vec{s})e}{m_b(\vec{n},\vec{s})e}}\left(\frac{K_i(\lambda)}{K_j(\lambda)}\right) = \left(\frac{K_i(\lambda)}{K_j(\lambda)}\right) \tag{3.5}$$

For shiny objects the effect of subtracting one band j to another i produces a highlight invariant representation (see Equation 3.6), since the second part of Equation 3.2 does not depend on the band:

$$\begin{aligned} C_i - C_j &= m_b(\vec{n},\vec{s})eK_i(\lambda) + \cancel{(m_s(\vec{n},\vec{s},\vec{v})ec_s f)} \\ &\quad - m_b(\vec{n},\vec{s})eK_j(\lambda) + \cancel{(m_s(\vec{n},\vec{s},\vec{v})ec_s f)} \\ &= m_b(\vec{n},\vec{s})e(K_i(\lambda) - K_j(\lambda)) \end{aligned} \tag{3.6}$$

For shiny object, first subtracting and second dividing produces an illumination intensity, object geometry and highlight invariant representation:

$$\begin{aligned} \frac{C_i - C_j}{C_k - C_l} &= \frac{m_b(\vec{n},\vec{s})e(K_i(\lambda) - K_j(\lambda))}{m_b(\vec{n},\vec{s})e(K_k(\lambda) - K_l(\lambda))} \\ &= \cancelto{1}{\frac{m_b(\vec{n},\vec{s})e}{m_b(\vec{n},\vec{s})e}}\left(\frac{K_i(\lambda) - K_j(\lambda)}{K_k(\lambda) - K_l(\lambda)}\right) = \left(\frac{K_i(\lambda) - K_j(\lambda)}{K_k(\lambda) - K_l(\lambda)}\right) \end{aligned} \tag{3.7}$$

Table 3.1 summarizes the main properties of some of the most important color invariant representations. They are the original chromatic RGB color representation, the rgb color representation (see Equation 3.8) and the one bands Hue H

	Object geometry, Viewing direction, illumination intensity	Highlights
RGB	×	×
rgb	✓	×
S	✓	×
H	✓	✓
$c_1c_2c_3$	✓	×
$l_1l_2l_3$	✓	✓

Table 3.1: *Resume of some of the main color invariants.* ✓ *stands for ”it is invariant to” and* × *stands for ”it is NOT invariant to”.*

(See Equation 3.9) and Saturation S (see Equation 3.10) representations. In addition, Gevers and Smeulders presented in [Gevers and Smeulders, 1999] two new invariant representations: $c_1c_2c_3$ (See Equation 3.11) and $l_1l_2l_3$ (See Equation 3.12), which have some advantages with respect to other invariant representations in object recognition problems.

$$\begin{cases} r(R,G,B) = \dfrac{R}{R+G+B} \\ g(R,G,B) = \dfrac{G}{R+G+B} \\ b(R,G,B) = \dfrac{B}{R+G+B} \end{cases} \tag{3.8}$$

$$H(R,G,B) = \arctan\left(\frac{\sqrt{3}(G-B)}{R-G)+(R-B)}\right) \tag{3.9}$$

$$S(R,G,B) = 1 - \frac{\min(R,G,B)}{R+G+B} \tag{3.10}$$

$$\begin{cases} c_1(R,G,B) = \arctan\left(\dfrac{R}{\max(G,B)}\right) \\ c_2(R,G,B) = \arctan\left(\dfrac{G}{\max(R,B)}\right) \\ c_3(R,G,B) = \arctan\left(\dfrac{B}{\max(R,G)}\right) \end{cases} \tag{3.11}$$

$$\begin{cases} l_1(R,G,B) = \left(\dfrac{(R-G)^2}{(R-G)^2+(R-B)^2+(G-B)^2}\right) \\ l_2(R,G,B) = \left(\dfrac{(R-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}\right) \\ l_3(R,G,B) = \left(\dfrac{(G-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}\right) \end{cases} \tag{3.12}$$

3.3 Spatially varying illumination model

Although the use of invariant representations can help in some problems, for instance, to improve the recognition rate in object recognition, it is preferible to include the illumination parameters into the objective function and therefore to jointly estimate the illumination parameters and the motion ones. This section deals with this approach. One of the main problems of the invariant color representations is that the invariants are based on assumptions that are often not very realistic in real world. Therefore, many times these assumptions fail and it is not possible to fulfill the aim of the invariant representation. In addition, they usually introduce noise to images and also destroy border and texture information which could be crucial for image registration. The use of an image model into the objective function to be minimized is a more desirable way to deal with illumination changes. That is the reason because this is the approach that has been used in the proposed global motion estimation technique presented in this Chapter.

Conventional intensity-based motion estimation methods are based on the brightness constancy assumption given as follows:

$$I_1(x_i, y_i) - I_2(x'_i, y'_i) = 0, (\forall i \in \Re), \tag{3.13}$$

where $I_1(x_i, y_i)$ is the gray level of the first image in the sequence (test image) at the point (x_i, y_i), and $I_2(x'_i, y'_i)$ is the gray level of the second image in the sequence (reference image) at the transformed point (x'_i, y'_i). $\Re$ is the region of interest.

Some preliminary works [Szeliski and Coughlan, 1997] used an illumination model to account for uniform photometric variation as follows:

$$\alpha I_1(x_i, y_i) + \beta - I_2(x'_i, y'_i) = 0, \tag{3.14}$$

where the constant α and β are the illumination multiplication and bias factor, respectively. This illumination model comes from the simplification of the

Shafer's model (see Equation 3.2) where the intensity of a pixel can be broadly explained as the product of the sensor by a factor (α) plus a bias factor (β). The main problem of that illumination model is that it cannot account for spatially varying illumination conditions, that is, it assumes that all the pixels have the same α and β. To overcome this restriction, a more general dynamic image model [Negahdaripour, 1998] can be used where the multiplication and bias factor are functions of localization, i.e. $\alpha \equiv \alpha(x_i, y_i)$ and $\beta \equiv \beta(x_i, y_i)$. Assuming that these two illumination factors are slowly varying functions of localization, they can be well approximated by low-order polynomials. For instance, $\alpha(x_i, y_i)$ and $\beta(x_i, y_i)$ can be expressed using bilinear and constant polynomials, respectively, as follows:

$$\begin{aligned} \alpha(x_i, y_i) &= \alpha_x x_i + \alpha_y y_i + \alpha_c \\ \beta(x_i, y_i) &= \beta_c \end{aligned} \tag{3.15}$$

Applying this Dynamic Image Model (DIM), Eq. (3.14) can be expressed using Eq. (3.15) as follows:

$$\alpha(x_i, y_i) I_1(x_i, y_i) + \beta(x_i, y_i) - I_2(x'_i, y'_i) = 0. \tag{3.16}$$

3.4 GLS-based global motion estimation under varying illumination

The GLS-based global motion estimator is a non-linear motion estimation technique proposed in this work as an alternative method to M-Estimators [Bober and Kittler, 1994b], [Odobez and Bouthemy, 1995] and other robust techniques to deal with outliers in motion estimation scenarios. Chapter 2 has been completely devoted to explain it.

In the original method, each $F_i(\chi, \lambda_i)$ was expressed as follows: $F_i(\chi, \lambda_i) = I_1(x_i, y_i) - I_2(x'_i, y'_i)$, i.e. the *BCA*. In this work, a dynamic image model which allows spatially varying illumination is used instead (see Equation 3.16). Therefore, each $F_i(\chi, \lambda_i)$ is expressed as follows:

$$F_i(\chi, \lambda_i) = \alpha(x_i, y_i) I_1(x_i, y_i) + \beta(x_i, y_i) - I_2(x'_i, y'_i). \tag{3.17}$$

Now, the vector of parameters χ depends on the motion and illumination models used. In this chapter, affine motion (6 parameters) and bilinear (3 parameters) and constant (1 parameter) polynomials for multiplication and bias factors (see Equation 3.15), respectively, have been used. Therefore, the vector of parameters is defined as:

$$\chi = (a_1, b_1, c_1, a_2, b_2, c_2, \alpha_x, \alpha_y, \alpha_c, \beta_c)^T. \tag{3.18}$$

In order to calculate the matrices A_i, B_i and E_i (see Equations 2.14, 2.15 and 2.16, respectively), the partial derivatives of the function $F_i(\chi, \lambda_i)$ with respect to the parameters and with respect to the observations must be worked out. The resulting A_i, B_i and E_i using affine motion are expressed as follows:

$$\begin{aligned} B_i &= (\alpha_x I_1 + \alpha_i I_1^x - a_1 I_2^x - a_2 I_2^y, \alpha_y I_1 + \alpha_i I_1^y - b_1 I_2^x - b_2 I_2^y, \alpha_i) \\ A_i &= (-x_i I_2^x, -y_i I_2^x, -I_2^x, -x_i I_2^y, -y_i I_2^y, -I_2^y, x_i I_1, y_i I_1, I_1, 1.0) \\ E_i &= -\left(\alpha_i I_1(x_i, y_i) + \beta_c - I_2(x'_i, y'_i)\right) \end{aligned} \tag{3.19}$$

where I_1^x, I_1^y, I_2^x and I_2^y have been introduced to simplify notation as:

$$\begin{aligned} I_1^x &= I_1^x(x_i, y_i) \\ I_1^y &= I_1^y(x_i, y_i) \\ I_2^x &= I_2^x(x'_i, y'_i) \\ I_2^y &= I_2^y(x'_i, y'_i) \end{aligned} \tag{3.20}$$

with $I_1^x(x_i, y_i)$ and $I_1^y(x_i, y_i)$ being the gradients of the test image at point (x_i, y_i); and with $I_2^x(x'_i, y'_i)$ and $I_2^y(x'_i, y'_i)$ being the gradients of the reference image at point (x'_i, y'_i). α_i has been also introduced as:

$$\alpha_i = \alpha(x_i, y_i). \tag{3.21}$$

3.4.1 Motion and illumination parameters initialization

In many motion estimation problems where the deformation between images is quite large (e.g. large rotation, strong changes of scale, etc.), it is necessary to initialize the motion estimator using a good initial vector of motion parameters. For this purpose, first the feature-based method explained in Chapter 2 is used to obtain the initial vector of parameters that are not very far from the true solution. Using this initialization (i.e. $\widehat{\chi}(0)$), in the second step, the GLS-based global motion estimator using the dynamic illumination model, is applied, which refines the estimation of the motion and illumination parameters up to the accuracy level desired by the user. Regarding the illumination parameters at $\widehat{\chi}(0)$, they have initially been set to: $\alpha_x = \alpha_y = \beta_c = 0$ and $\alpha_c = 1$.

3.5 Experimental results

In this section, a set of motion estimation experiments are performed in order to test the accuracy of the proposed technique. In particular, the accuracy of the estimation in the case of using the Brightness Constancy Assumption (*BCA*, see Equation 3.13) is compared with the case when the Dynamic Image Model (*DIM*, see Equation 3.16) is used instead. To check the accuracy of the estimation, the normalized correlation coefficient (Ncc) similarity measure has been calculated using the pixels of the overlapped area of both images. The absolute value of Ncc gives values from 0.0 (low similarity) to 1.0 (high similarity), and is expressed as follows:

$$Ncc(I_1, I_2) = \frac{\sum_{(x_i,y_i)\in\Re}[(\alpha_i I_1 + \beta_i - \mu_1)(I_2 - \mu_2)]}{\sqrt{\sum_{(x_i,y_i)\in\Re}(\alpha_i I_1 + \beta_i - \mu_1)^2 \sum_{(x_i,y_i)\in\Re}(I_2 - \mu_2)^2}}, \tag{3.22}$$

where μ_1,μ_2 are the average of the gray level of both images, $\Re$ the overlapped area and I_1, I_2, α_i and β_i have been introduced to simplify notation as: $I_1 \equiv I_1(x_i, y_i)$, $I_2 \equiv I_2(x'_i, y'_i)$, $\alpha_i \equiv \alpha(x_i, y_i)$ and $\beta_i \equiv \beta(x_i, y_i)$.

A set of challenging sets of image pairs have been selected. They can be downloaded from Oxford's Visual Geometry Group web page [1] except for the last set that has been obtained from Internet. Oxford's ones present three main types of changes between images in 4 different sets of images; Blur: *Bikes* set, global illumination: *Leuven* set and zoom+rotation: *Bark* and *Boat* sets. Each image set has 6 different images. A sample of the images are showed in the appendix A. For each set, the 6 images have been combined in all possible pairs ($1 \leftrightarrow 2$, $1 \leftrightarrow 3$, …, $1 \leftrightarrow 6$, $2 \leftrightarrow 3$, …, $5 \leftrightarrow 6$). The *satellite* set is a set of images from the same area but they have been captured at different times/days and therefore with different illumination conditions.

To introduce a large illumination variation in the data, the second image of each image pair $I_1 \leftrightarrow I_2$ is modified multiplying it by a multiplier function. Two multipliers have been used, the first one makes dark the image from left to right and the second one has the form of a Gaussian. They are showed at the first row of Figure 3.3. The second row of Figure 3.3 shows an example of application of the multipliers. The resulting images, after the application of the multipliers, are called I_2^{Gd} and I_2^{Gn}, respectively. Note that the illumination changes of *Leuven* set (see Figure A.5 are different from the ones introduced by the multipliers, since, in the first case, the changes are global, i.e. the changes do not depend on

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

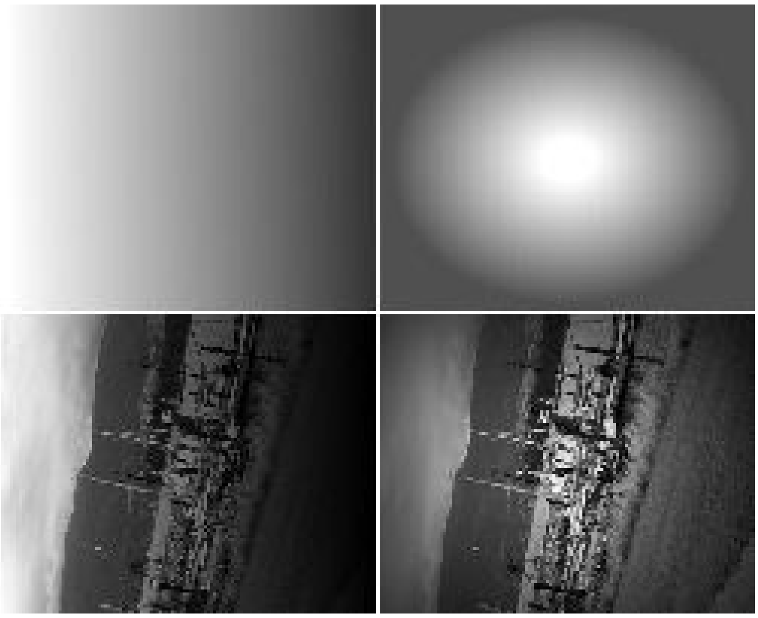


Figure 3.3: *The first row shows the multipliers used to add large illumination variation to the data. The second row shows an example of the resulting images after applying the multipliers.*

the localization of the pixel, while, in the second case, the multipliers introduces illumination changes which depend on the localization of the pixel.

For each image pair: $I_1 \leftrightarrow I_2$, the proposed motion estimation techniques is applied in order to obtain six Ncc values. First, the proposed motion estimation technique is performed using the original images (i.e. I_1 and I_2) with the BCA and the DIM to obtain two Ncc values: $Ncc(BCA)$ and $Ncc(DIM)$. In the second step, the image I_2^{Gd} is used as second image, producing the Ncc values: $Ncc^{Gd}(BCA)$ and $Ncc^{Gd}(DIM)$. Finally, the same process is repeated using now the image I_2^{Gn} obtaining the Ncc values: $Ncc^{Gn}(BCA)$ and $Ncc^{Gn}(DIM)$.

Figures 3.4, 3.5 and 3.6 show the mean of the Ncc obtained for each set, when original image I_2 (i.e. image pair $I_1 \leftrightarrow I_2$), modified image I_2^{Gd} (i.e. image pair $I_1 \leftrightarrow I_2^{Gd}$) and modified image I_2^{Gn} (i.e. image pair $I_1 \leftrightarrow I_2^{Gn}$) are used as second image, respectively.

In general, the use of the dynamic image model instead of the BCA provides more accurate results in almost all the cases. Figure 3.4 shows that although non additional illumination changes have been artificial added, the use of the dynamic image model improves the accuracy of the estimation, since, probably, there is

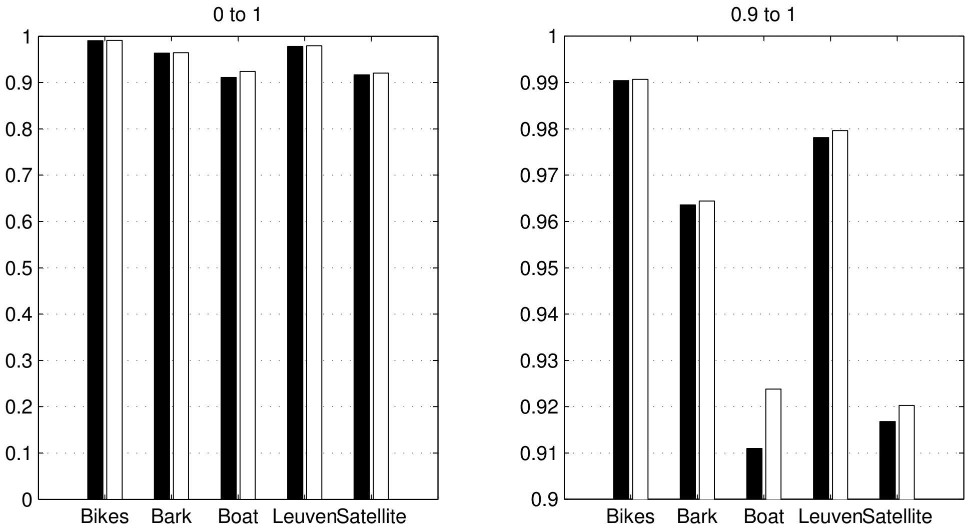


Figure 3.4: *Results obtained when the images I_2 are used as second image. The right graphic shows the results in* $[0.9, 1]$ *range. Black: BCA-based; White: DIM-based.*

a small (just no visually appreciable, but existing) illumination variation due to the acquisition process. The accuracy level is very similar in both cases, but in most cases, the use of the DIM improve the accuracy of the estimation. Note that at *Leuven* set, the GLS-based motion estimator obtains accurate estimates even when the *BCA* is used, that is due to the weights used at the estimation procedure depend on gradient information and not on the grey level (see Chapter 2 for details).

Figures 3.5 and 3.6 show how the accuracy of the estimation is drastically reduced when using the *BCA*, since the strong illumination changes introduced make that the *BCA* is not fulfilled at the majority of the observations and, therefore, the estimation procedure gets lost while searching for the optimal parameters in the minimization process. The dynamic image model can deal with this situation, and therefore, when using it, the accuracy of the estimation is improved. Note how the accuracy level obtained when using the first multiplier (i.e. using I_2^{Gd} as the second image) is as good, and even better, as the cases when no illumination changes have been introduced to second image.

The second multiplier (i.e. using I_2^{Gn} as second image) introduces stronger illumination changes than the first one, since the illumination changes introduced can not perfectly been modelled using the dynamic model proposed. Therefore,

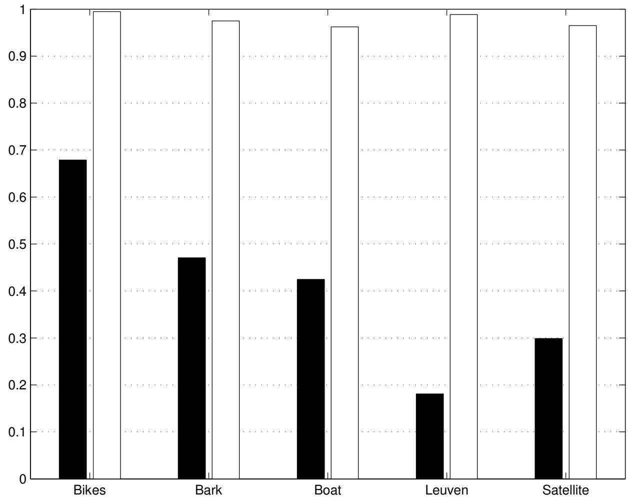


Figure 3.5: *Results obtained when the images $I_2^{Gd}$are used as second image. Black: BCA-based; White: DIM-based.*

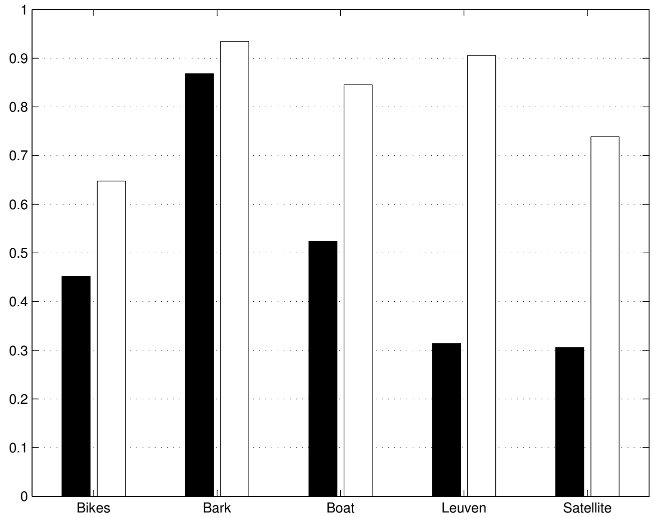


Figure 3.6: *Results obtained when the images I_2^{Gn} are used as second image. Black: BCA-based; White: DIM-based.*

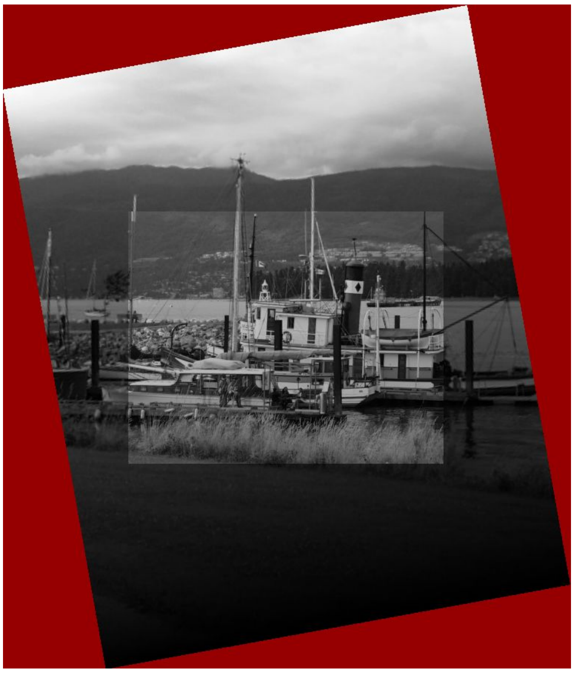


Figure 3.7: *Mosaic image created using as input images from the Boat set.*

the accuracy obtained is not as good as in the previous case, but it still maintains high accuracy levels.

In order to show an illustrative example of the behavior of the proposed approach, a mosaic image has been created using the motion parameters obtained from the motion estimation experiment, using as input images the first image from *Boat* image set as I_1, and the left image of the second row of Figure 3.3 as I_2 (i.e. the 4th image from *Boat* image set after adding illumination noise). In despite of the strong illumination differences between both inputs images, the proposed techniques obtains accurate estimates as can be showed at Figure 3.7.

In order to see how the illumination parameters have also been accurately estimated, the second image of the previous experiment has been transformed using the motion and illumination parameters obtained with the proposed approach. That image (called I_1') should be quite similar to the first image of the pair if the parameters have been accurately estimated. Figure 3.8 shows both images and the grey level differences between both images. In spite of both input images have been captured at different time moments, and therefore, the vegetation, people, water and even the boat are not completely stationary between both

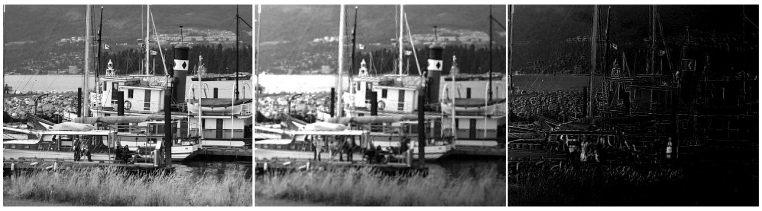


Figure 3.8: *Original I_1 (left), estimated I_1' (middle) and grey level difference (right).*

images, the images show that the illumination parameters have been estimated with high accuracy.

3.6 Conclusions

In this chapter, the accurate Generalized least squared-based global motion estimator presented in Chapter 2 has been used in combination with a dynamic image model where the multiplication and bias illumination factors are functions of the localization (x, y). Experiments using challenging real images have been performed to show that with the combination of both techniques, a global motion estimator can be obtained, which can perform the motion estimation task in an accurate manner while allowing large deformation and illumination changes between images.

Chapter 4

Quasi simultaneous motion estimation and segmentation

Contents

4.1 Introduction 82

4.2 Terminology used in this chapter 84

4.3 Quasi-simultaneous parametric multimodel fitting . . 85

4.4 The proposed algorithm 87

4.5 Experimental results 99

4.6 Conclusions 102

This chapter presents a new framework for the motion segmentation and estimation task on sequences of two grey images without a priori information of the number of moving regions present in the sequence. The proposed algorithm uses temporal information, by using the accurate generalized least-squares global motion estimation process (explained in Chapter 2) and spatial information by using an iterative region growing algorithm which classifies regions of pixels into the different motion models present in the sequence. The initial regions of pixels are obtained from a given grey-level segmentation process. The performance of the algorithm is tested on synthetic and real images with multiple objects undergoing different types of motion.

4.1 Introduction

Segmentation of moving objects in a video sequence is basic task for several applications in computer vision, e.g. a video monitoring system, intelligent-highway system, tracking, airport safety, surveillance tasks and so on. In this chapter, motion segmentation, also called spatial-temporal segmentation, refers to labelling pixels which are associated with different coherently moving objects or regions in a sequence of two images. Motion estimation refers to assigning a motion vector to each region in an image. In this case, the motion estimation problem is neither 100% global nor 100% local. It can be considered as semi-global in the sense that the motion is estimated for all the pixel of the region of interest, which is composed of several pixels, but never as big as the complete image and as small as a single pixel either.

Performing motion estimation and motion segmentation simultaneously usually falls in a *Hen-and-egg* problem. It is due to the fact that data classification and parameter estimation strongly depend on each other. It is known that, on the one hand, if the data is well-classified, i.e, we know which pixel support which model, then it is easy to obtain accurate estimates for the parameters. On the other hand, if we know accurate estimates of the parameters, then it is straightforward to classify the pixels into the models.

The motion segmentation and estimation problem has been formulated in many different ways ([Irani et al., 1994], [Odone et al., 2000], [Kim and Kim, 2003], [J. et al., 2001], [Bad-Hadiashar et al., 2002], [Ayer et al., 1994], [Bober and Kittler, 1993]). We choose to approach this problem as a multi-structural parametric fitting problem. In this context, the segmentation problem is similar to robust statistical regression. The main difference is that robust statistical regression usually involves statistics for data having one target distribution and corrupted with random outliers. Motion segmentation problems usually have more than one population with distinct distributions and not necessarily with a population having absolute majority.

The problem of fitting an a priori known model to a set of noisy data (with random outliers) has been studied in the statistical community for a number of decades. One important contribution was the Least Median of Squares (LMedS) robust estimator [Rousseeuw, 1984] but it has the break down point of 50%. This means that LMedS technique needs the population recovered to have at least a majority of 50% (plus 1). Other robust estimators have been developed in order to overcome this problem, which is frequently encountered in different computer vision tasks. They are Adaptive Least k-th Order residual (ALKS) [Lee et al., 1998] and Minimum Unbiased Scale Estimator (MUSE) [Miller and

Stewart, 1996]. These techniques minimize the k-th order statistic of the square residuals where the optimum value for the k is determined from the data. The problem of both techniques is the estimation of the correct value of k suffers high computation effort.

To overcome the computational complexity, Bab-Hadiashar and Suter presented a method named Selective Statistical Estimator (SSE) [Bad-Hadiashar et al., 2002], which is a variation of the Least K-th order statistic data regression where the user proposes the value k as the lower limit of the size populations one is interested in. All the LKS-based algorithms start selecting an initial model using random sampling, and classifying all the observations into this model using a scale measure. With the remaining observations the process is repeated until all the observations have been classified. The main problem of these algorithms is that there are frequently observations that can be more suitable to belong to a model but they have been classified in an earlier model.

Danuser and Stricker [Danuser and Stricker, 1998] presented a similar framework for parametric model fitting. Their algorithm has a fitting step that is one component of the algorithm that also collects model inliers, detects data outliers and determines the a priori unknown total number of meaningful models in the data. They apply a quasi simultaneous application of a generalized least squares fitting while classifying observations in the different parametric data models. They applied their algorithm to multiple lines and planes fitting tasks. The most important advantages with respect to LKS-based algorithms are the use of an exchange step, that permits change of observations among models, and the use of a inliers/outliers classification process, which increases the accuracy of the segmentation. The Danuser and Stricker algorithm will be presented in more detail in Section 4.3 since some of their ideas have been used in the proposed approach.

In [Montoliu and Pla, 2001a] a quasi-simultaneous motion segmentation and estimation method based on a parametric model fitting algorithm was presented. The method accurately estimates the affine motion parameters using a generalized least squares fitting process. It also classifies the pixels into the motion models present in two consecutive frames. This algorithm uses each pixel of the image as observation. It suffers from problems of isolated points because it does not use spatial neighborhood information and need good initial models to obtain the final motion segmentation. Nevertheless, it indicates that the quasi-simultaneous application of the inliers/outliers classification algorithm and the accurate motion estimator can be useful to be applied in motion segmentation tasks.

In this chapter, a motion segmentation and estimation algorithm that, instead of using the pixel as observation, it uses regions of pixels, is presented. The use of regions made the segmentation more spatially consistent. In addition, the algorithm uses neighborhood constraints to collect new inliers to the model, only regions that are neighbor of the model are considered to be inliers. This algorithm also overcomes the need of a previous good segmentation of the models, and allows extracting the model without a priori information of the number of moving regions present in the sequence.

The proposed algorithm has been designed to be applied to general purpose motion segmentation problems, without a priori information of the origin of the images. In more specific problems, the knowledge of some properties of the scene can help to obtain accurate segmentation. For instance, in traffic scenes the background (the road) usually is static and therefore can be removed, simplifying the segmentation process. However, this assumption cannot always be made in other problems. For this reason, our algorithm has been designed to be applied to all kind of motion segmentation problems. No specific information about the scenes, like the existence of static regions, the size and the shape of the objects, the motion of the sensor used to captured the images, etc., is given.

Summarizing, the main characteristics of the proposed approach are the followings:

- A GLS Motion Estimation algorithm is used, which produces accurate estimation of the motion parameters.
- The classification process collects inliers, rejects outliers and exchanges regions among models allows to improve motion segmentation.
- It uses regions of pixels instead of pixels as observations and neighbor information, that improves the spatial consistency.

The rest of the chapter is organized as follows: the next section summarize the terminology used in the chapter. Section 4.3 presents a brief explanation of the Danuser and Stricker's algorithm. Section 4.4 explains the proposed motion segmentation and estimation algorithm. Section 4.5 presents a set of experiments in order to verify the results obtained with our approach. Finally, in the last section, some conclusions drawn from this work are described.

4.2 Terminology used in this chapter

In this chapter the following terminology is used:

- **Model** as a structure with two elements, the first is a parametric motion vector χ and the second is a set of observations Φ of the image that support the model.

- **Region** as a set of pixels with grey-level coherence.
- **Inlier** as an observation that supports the motion of a model, i.e. it has a very high likelihood of performing the motion of the model.
- **Outlier** as an observation that does not support the motion of a model, i.e. it has a very low likelihood of performing the motion of the model.
- $\Re$ as the set of all observations.
- Ω as the set of not yet classified observations.
- M_j as the present model.
- Υ as the set of extracted models. $\Upsilon = [M_1, \ldots, M_n]$.
- $\emptyset$ as the empty set.

4.3 Quasi-simultaneous parametric multimodel fitting

The proposed motion segmentation and estimation algorithm has been designed following some ideas from Danuser and Stricker's framework for parametric multi model fitting. Therefore, it could be useful to explain with more detail this technique. This section deals with it. The reader is referenced to [Danuser and Stricker, 1998] for a comprehensive study of the technique which has been successfully applied to line and plane fitting problems.

The Figure 4.1 shows the Danuser and Stricker's algorithm to extract multiple models from a set of observations points $\Re$. For instance, each observation can be an individual point 2D (for line fitting problems) or 3D (for plane fitting problems). There are three main functions in this framework. They are briefly explained as follows:

- **InitialModelDetection**(Ω): The objective of this function is to detect an initial model. That is, the objective is to detect an initial set of observations from Ω (the complete set of not yet classified observations) that have a certain grade of probability of belonging to the same model. This can be done using a random sampling technique as *RANSAC* or similar. It is assumed that the initial model has outliers and also that not all the observations that really are inliers of this model have been included in the initial model detected.

```
Input: A set of unclassified observation points $\Re$
Output: A set $\Upsilon$ of valid models $\Upsilon = [M_1, \ldots, M_n]$, with $M_i = [\Phi_i, \chi_i]$.
$\Omega \leftarrow \Re$
$\Upsilon \leftarrow \emptyset$
$[M_1, \Omega] \leftarrow$ InitialModelDetection($\Omega$)
$j \leftarrow 1$
while (An initial $M_j$ has been detected) do
    $M_j \leftarrow$ SingleModelExtraction($M_j, \Omega$)
    if $M_j$ is valid then
        $\Upsilon \leftarrow$ ExchangePointsBetweenModels($M_j, \Upsilon$)
        $\Upsilon \leftarrow \Upsilon \cup M_j$
    else
        $\Omega \leftarrow \Omega \cup \Phi_j$
        delete $M_j$
    end if
    $j \leftarrow j + 1$
    $[M_j, \Omega] \leftarrow$ InitialModelDetection($\Omega$)
end while
Refit all the models.
```

Figure 4.1: *Danuser and Stricker's multiple model extraction algorithm.*

- **SingleModelExtraction**(M_j, Ω): The steps of this function are summarized in the algorithm showed in Figure 4.2. The input is a detected initial model. The process first looks for outliers in the observations belonging to the present model. Those observations are deleted from the model and included in the pull Ω. Then, the parameters are estimated to improve the estimation. The next step is to look for inliers in order to check if, with the improved parameters, there exist observations in Ω that now could be considered as inliers. This loop is repeated until no new observations are deleted or inserted into the model. The detection of outliers and inliers are performed using two statistical tests. In addition, this framework also include an statistical test to check if the current model whether is valid or not.

- **ExchangePointsBetweenModels**(M_j, Υ): The data exchange between the models is the key to make the estimation results independent of the order in which the models are extracted. The data exchange is carried out between the last extracted model M_j and all previously extracted models $[M_1, \ldots, M_{j-1}]$. For each observation from a model previously extracted, an statistical test is performed to check if this observation can be introduced in the new extracted model M_j. In this case, the observation is moved from the model where it was previously located to the last extracted one.

The statistical tests used to detect inliers and outliers are based on the data snooping technique [Baarda, 1968]. The goal of the data snooping technique is to search and eliminate observations which are perturbed by gross errors. This concept is similar to the M-Estimators with the difference that an observation with a significantly large residual has absolute no influence on the parameter estimation and that the necessary classification of the residuals is made based on a statistical test.

4.4 The proposed quasi simultaneous motion estimation and segmentation algorithm

The inputs of the algorithm are two images of a sequence, the first one I_1 (called reference image) captured at time t and the second one I_2 (called test image) captured at time $t+1$. The outputs of the algorithm are a motion-based segmentated image I_s and a list of models $\Upsilon = [M_1, \ldots, M_n]$, where each M_i is composed of a vector of motion parameters χ_i and a set of regions Φ_i that support the model. All the pixels belonging to the regions of Φ_i are labelled using the same color in I_s.

4.4.1 Motion parameters estimation for a model

The proposed Generalized Least Squares-based (GLS) motion estimation technique is used in order to obtain the motion parameters of a model (see Chapter 2).

A model M_j has two elements, the motion parameters χ_j and the set of regions that support the model $\Phi_j = [R_1, \ldots, R_N]$, with N being the number of regions in Φ_j.

For estimating the motion of the model, first it is necessary to clarify, following the terminology presented in Chapter 2, which are the vector of input observations, i.e. λ. In this case, the vector of observations is made up of all pixels that belong to each region of the set Φ_j. In this case, we prefer to use the

```
Input: An initial model detected $M_j = [\Phi_j, \chi_j]$ and the set $\Omega$ of not
    yet classified observations.
Output: An improved model $M_j$ and the set $\Omega$ actualized.
exit ← false
while exit=false do
    outs ← LookForOutliers($\Phi_j$)
    $\Phi_j \leftarrow \Phi_j - \text{outs}$
    $\Omega \leftarrow \Omega \cup \text{outs}$
    $\chi_j \leftarrow \text{Fit}(\Phi_j)$
    ins ← LookForInliers($\Omega$)
    $\Omega \leftarrow \Omega - \text{ins}$
    $\Phi_j \leftarrow \Phi_j \cup \text{outs}$
    if there is not any change in $\Phi_j$ then
        $M_j$ is a valid model.
        exit ← true.
    else
        $\chi_j \leftarrow \text{Fit}(\Phi_j)$.
        if $M_j$ is not a valid model then
            $M_j$ is not a valid model.
            exit ← true.
        end if
    end if
end while
```

Figure 4.2: *Danuser and Stricker's single model extraction algorithm.*

notation λ_{M_j} as the set of observations of the model M_j. It can be expressed as follows:

$$\lambda_{M_j} = \{\lambda_i = (x_i, y_i, I_1(x_i, y_i))/(x_i, y_i) \in R_k; R_k \in \Phi_j\} \tag{4.1}$$

where (x_i, y_i) are the pixel coordinates of the observation λ_i (see Equation 2.20) and $k = [1, \ldots, N]$.

Thus, the motion of the model is estimated using the procedure explained in subsection 2.4.3 using λ_{M_j} as the set of observations.

```
**Input:** Two input images $I_1$ and $I_2$
**Output:** A set $\Upsilon$ of valid models $\Upsilon = [M_1, \ldots, M_n]$, with $M_i = [\Phi_i, \chi_i]$.
$\Re \leftarrow$ SegmentImageInRegions($I_2$)
$\Omega \leftarrow \Re$
$\Gamma \leftarrow \emptyset$
$\Upsilon \leftarrow \emptyset$
$G \leftarrow$ CreateAdjacencyGraph($\Re, I_2$)
$M_1 \leftarrow$ GetInitialModel($I1, I2, \Omega, G$)
$j \leftarrow 1$
while (An initial $M_j$ has been detected) do
    $M_j \leftarrow$ ImproveModel($M_j, \Omega, I_1, I_2, G$)
    if $M_j$ is valid then
        $\Upsilon \leftarrow$ ExchangeOfRegions($I1, I2, M_j, \Upsilon, G$)
        $\Upsilon \leftarrow \Upsilon \cup M_j$
    else
        $\Gamma \leftarrow \Gamma \cup \Phi_j$
    end if
    $j \leftarrow j + 1$
    $M_j \leftarrow$ GetInitialModel($I1, I2, \Omega, G$)
end while
$\Upsilon \leftarrow$ FinalStep($I1, I2, \Upsilon, \Gamma, G$)
```

Figure 4.3: *Our proposed quasi-simultaneous motion estimation and segmentation algorithm.*

4.4.2 Algorithm outline

For the sake of clarity, we describe the proposed algorithm in 6 steps which are summarized in the algorithm showed at Figure 4.3. The 6 steps are the followings:

1. **Preliminaries**: In this step, I_2 is segmented using a given grey level segmentation algorithm. The regions obtained are used as input of the algorithm. An adjacency graph of the previous segmentation is created. In addition, the spatial derivatives of the images I_1 and I_2 are estimated.

   The purpose of the grey-level segmentation process is to classify the pixels into regions. Our motion segmentation algorithm requires that each segmented region should not have pixels belonging to more than one final motion models. Any grey level segmentation algorithm that fulfill the

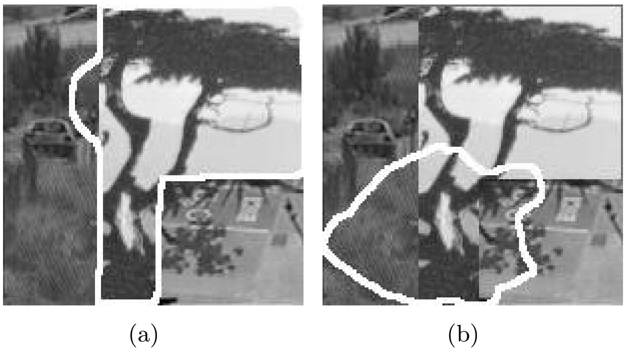


Figure 4.4: *Two examples of initial models.*

previous constraint can be used. A sieve-based grey level segmentation algorithm [Bangham et al., 1998] has been used, since it produces a hierarchical representation of the image with different segmentations that differ in region size. A segmentation with small regions must be used to fulfill the constraint.

2. **Get Initial Model**: The aim of this process is find the best possible start point to the global motion segmentation and estimation algorithm. A good initial model is made up of a set of regions that have a high likelihood to belong to the same model. The process starts selecting a region randomly. A model with this region and its neighbors is formed. The motion is estimated for this model using the process introduced in subsection 4.4.1.

   A goodness measure GM is calculated for this model. This step is repeated q times. The model with the best goodness measure is selected as the initial model. The goodness measure is calculated using the following expression:

$$GM = ((1 - l_{avg}) * 2 + (l_{best} - l_{worst})), \tag{4.2}$$

   where l_{avg} is the average of the likelihood $L_{M_j}(R)$ for each region R using the motion model M_j, l_{best} is the highest likelihood of the regions and l_{worst} is the lowest likelihood of the regions. Therefore, the best initial model is the one which has the less GM.

   $L_{M_j}(R)$ is expressed as follows:

$$
\begin{aligned}
L_{M_j}(R) &= (\sum_{p_i \in R} L_{M_j}(p_i))/N_R, \\
L_{M_j}(p_i) &= e^{\frac{-1}{2} * \frac{F^2_{M_j}(p_i)}{\sigma 2}},
\end{aligned}
\tag{4.3}
$$

where N_R is the number of pixels of the region R. For each pixel p_i belonging to the region R, the likelihood $L_{M_j}(p_i)$ of the pixel belonging to a model M_j is calculated. This likelihood ([Bober and Kittler, 1994a]) has been modelled as a gaussian like function where $F_{M_j}(p_i)$ is the residual for the pixel p_i of the objective function using the motion parametric vector of the model M_j. That is, $F_{M_j}(p_i)$ is expressed as follows:

$$
F_{M_j}(p_i) = I_1(x_i, y_i) - I_2(x'_i, y'_i), \tag{4.4}
$$

where (x_i, y_i) are the coordinates of the pixel p_i and (x'_i, y'_i) are calculated using the estimated motion parameters χ_j for the model M_j.

Figure 4.4 shows an illustrative example of two possible initial models for a sequence with three different motion models: static (left part of the image) and two translational motions (the part of the image showing a tree and the bottom right part). The limits of two possible initial models are drawn with a continuous white line. Note that in the left image (Figure 4.4a) the majority of the pixels perform the same motion (the model of the three) and only a small area performs a different motion. Therefore, its GM will have a very small value. In addition, its GM will be lower than in the case of the right image (Figure 4.4b) where there is not a majority of pixels performing the same motion.

3. **Improve the model**: After an initial model has been obtained, an iterative classification process (It will be described with more detail in subsection 4.4.3) is started in order to find the inliers and to reject outliers between the regions that make up the initial model. The Figure 4.5 shows the algorithm of this process. With the set of resulting regions, we start another classification process with the neighbors of the last inserted regions not yet processed. This classification step continues until there are not more new neighbor regions to be processed. This algorithm is showed in Figure 4.6.

4. **Exchange of regions**: If a valid model M_j has been extracted, then a region exchange procedure is started. The goal of this procedure is to

reclassify regions that have been captured by an early model M_k where $k < j$. A region is moved if it lies closer to the new extracted model and there is a neighbor relationship between the region and the new model. If all the regions of the model M_k lie closer to the new Model M_j then the model M_k is deleted. When for each region of model M_k we can not decide if it lies closer to the model M_k or to the model M_j, then the models are merged, that is, it is considered both models have similar motion parameters.

5. **Repeat**: Go to step 2 and repeat the same process with another initial model if any. If there is any problem estimating the motion of some model, e.g. not enough texture information, not enough number of observations, etc., the regions of this model are moved to a set Γ called *regions with problems*.

6. **End**: When all possible models have been extracted, the models that only have one region are tested in order to try to merge them with their neighbor models. In addition, for each region in the Γ set is tested in order to move it into some of the models in its neighborhood.

At the end of the algorithm, a set Υ of motion models have been extracted. Each motion model is made up of a vector of motion parameters χ and a set of regions Φ which support the motion.

4.4.3 Inliers/Outliers region classification

The aim of this process is to classify the regions of a model (according to its motion parameters) in two sets, inliers: regions that support the motion parameters and outliers: regions that do not support them. The loop of this classification process consists of:

1. Estimate the motion parameters using all the pixels belonging to the regions of the model (see subsection 4.4.1).

2. Look for outliers into the regions of the model, if there are outliers, improve the motion parameters using only the remaining regions. A region R is considered outlier (with respect to model M_n) if the likelihood of region R belonging to a model M_j is lower than a threshold.

3. Test each outlier if it can be now considered inlier according to the new estimated parameters. If there are new inliers, the parameters are improved again. A region R is considered inlier (with respect to model M_j) if the

```
Input: A model $M_j = [\Phi_j, \chi_j]$, $\Phi_j = [R_1, \ldots, R_n]$, the input images
    $I_1$ and $I_2$ and the set of not yet classified observations $\Omega$.
Output: The improved model $M_j$. The set $\Omega$ is also modified.
$\Omega_{aux} \leftarrow \emptyset$
repeat
    $\chi_j \leftarrow \mathrm{Fit}(\Phi_j, I_1, I_2)$
    for all $R_i \in \Phi_j$ do
        if $R_i$ is outlier then
            $\Phi_j \leftarrow \Phi_j - R_i$
            $\Omega_{aux} \leftarrow \Omega_{aux} \cup R_i$
        end if
    end for
    $\chi_j \leftarrow \mathrm{Fit}(\Phi_j, I_1, I_2)$
    for all $R_i \in \Omega_{aux}$ do
        if $R_i$ is inlier then
            $\Phi_j \leftarrow \Phi_j \cup R_i$
            $\Omega_{aux} \leftarrow \Omega_{aux} - R_i$
        end if
    end for
until There are not changes in $\Phi_j$
$\Omega \leftarrow \Omega \cup \Omega_{aux}$
```

Figure 4.5: *Inliers/Outliers region classification algorithm.*

likelihood of the region R belonging to a model M_j is higher than a threshold.

4. Go to step 2 and repeat until there are not changes in the set of regions of the model.

In order to estimate a likelihood of a region R belonging to a model M_j, the expression $L_{M_j}(R)$ is used (see Equation 4.3). A region is considered as inlier when this measure is higher than a threshold and it is considered as outlier when its measure is lower than a threshold.

4.4.4 An illustrative example of the behavior of the algorithm

This subsection shows an illustrative example of how the proposed algorithm works. The input images are presented in Figure 4.7. Three different models can

```
Input: A model $M_j = [\Phi_j, \chi_j]$, the input images $I_1$ and $I_2$, the
    adjacency graph $G$ and the set of not yet classified observations
    $\Omega$
Output: The improved model $M_j$. The set $\Omega$ is also modified.
repeat
    $NH = [h_1, \ldots, h_m]$ such that $nh_i \in \Omega$ and it is neighbor of $R_j$,
    with $R_j \in \Phi_j$
    for all $h_i \in NH$ do
        if $h_i$ is inlier then
            $\Phi_j \leftarrow \Phi_j \cup h_i$
            $\Omega \leftarrow \Omega - h_i$
        end if
    end for
    $\chi_j \leftarrow \mathrm{Fit}(\Phi_j, I_1, I_2)$
until None of $h_i \in NH$ are included as inlier
```

Figure 4.6: *Algorithm to include new neighbor regions to a model.*

be found in this synthetic sequence. The first one corresponds to the left part of the image and stay static. The second one is the part of the image showing a tree. It performs a short translational motion. Finally, the third part of the image (the right-bottom corner) performs also a short translational motion but different from the previous one.

The first step is to segment the image into regions. The hypothetic results of this process is showed in the Figure 4.8. The left image shows each region of the image labelled using a different RGB color. The right one shows the label of each region that will be useful to explain the rest of the process. In addition, an adjacency graph must to be created to obtain the neighborhood relationships among regions. The Figure 4.9 shows the graph created for this example. The final result of the process must group the regions in three different models. The first one will group the regions $[1, 2, 3, 4]$, the second one will group the regions $[5, 6, 7, 8, 9]$ and finally the last one the regions $[10, 11, 12, 13]$.

The second step of the algorithm consists of extracting an initial model. Figure 4.10 shows two posible initial models. The first one includes the regions $[3, 5, 6, 7, 8]$ and the second one the regions $[2, 3, 4, 6, 7, 8, 9, 10, 12]$. Note how in the first one the majority of the regions belong to a valid final model, while in the second case, the are regions of the three final models. Therefore, the first initial model has a best goodness measure (see Equation 4.2) and then it is selected as

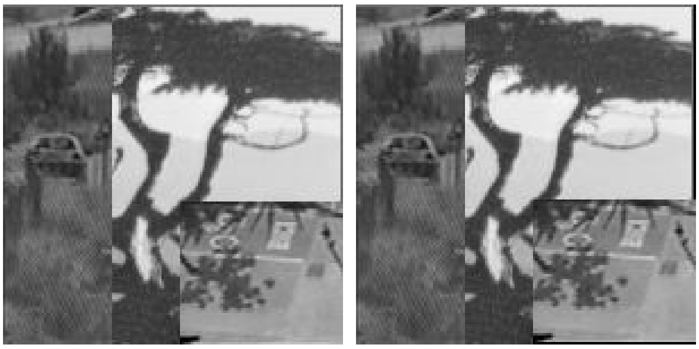


Figure 4.7: *Input images. Three model are present in this synthetic sequence. The left part stays static. The other two models (the tree one and the right-bottom corner) performs a different translational motion.*

the initial model to start the classification process.

The third step consists of improving the model. For this purpose, first a classification process is started with the regions of the initial model. In this example the model have the regions [3, 5, 6, 7, 8]. For each region, a test is performed to check if these regions can be considered as outliers. In this example, the test says that the region 3 is an outlier. The Figure 4.11 (left) show in red that region. Once the region 3 has been deleted from the model, no region is included or extracted from it. Then, now a new classification process is started to add new neighbor regions. The candidate regions to be included are: [1, 3, 2, 4, 10, 11, 12]. Only the region 9 can be considered as inliers. Once this region is included in the model, no region is included in the model. Then, the process of extraction the first model can be finished. The final model is showed in Figure 4.11 (right).

To explain the behavior of the fourth step, i.e. the exchange of regions between models, it can be useful to imagine that in the present situation of the algorithm two models have been extracted. The first one with the regions [5, 5, 7, 8, 10] and the second one with the regions [9, 11, 12, 13], that is the regions 10 and 9 have been bad classified. Then, the objective of this step is to detect this fact and therefore to move the region 10 to the second model and the region 9 to the first one. Figure 4.12 (left) illustrates this circumstance. Figure 4.12 (right) shows the final results after this step.

At the end of loop, three models should be extracted. In this example, since, there are not regions in the Γ set (regions with problems) and also there are not regions with only one region, the sixth step is not needed and therefore the

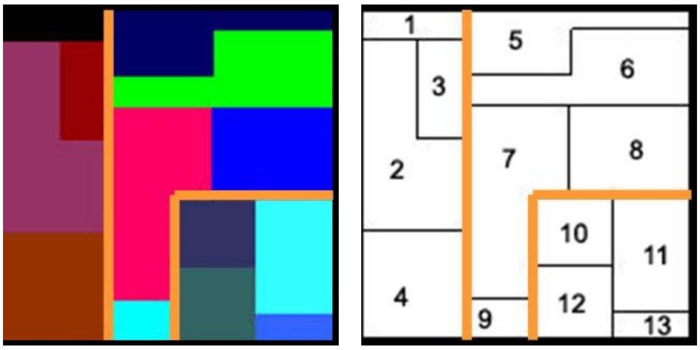


Figure 4.8: *Hypothetic results of the segmentation process (left) and labels of each region (right).*

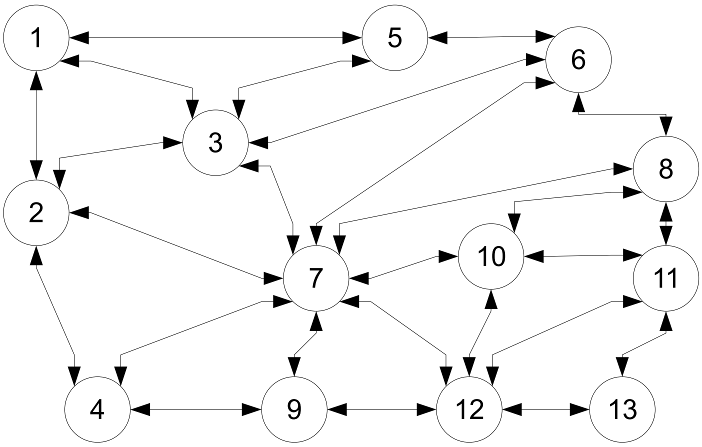


Figure 4.9: *Adjacency Graph created for the example. This graph shows the neighborhood relationships among regions.*

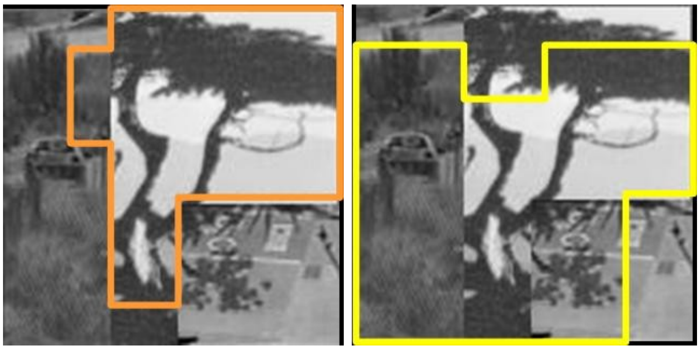


Figure 4.10: *Two possible initial models. The left one is a more convenient candidate to be selected.*

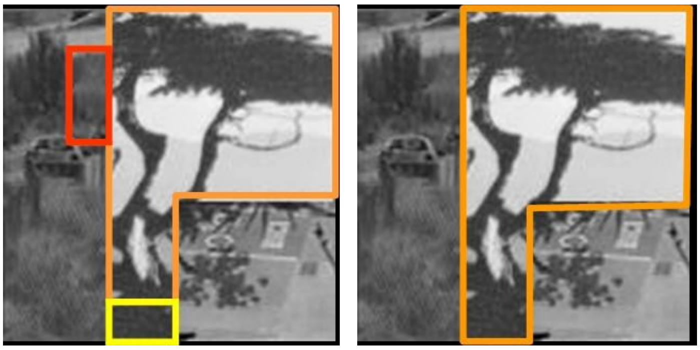


Figure 4.11: *Left: the red region is a candidate region to be deleted from the model, the yellow one is a candidate neighbor to be included in the model. Right: the results after the first model has been extracted.*

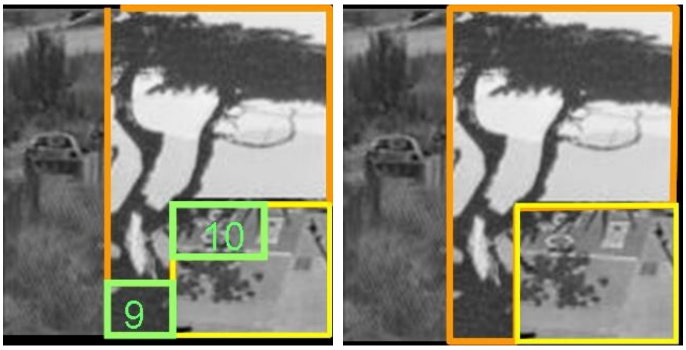


Figure 4.12: *Left: the regions 9 and 10 have been bad classified. Right: results after the exchange of region process has finished*

algorithm finishes.

4.4.5 Refining segmentation

The proposed motion segmentation approach requires that each region from the given grey-level segmentation should not have pixels belonging to more than one final motion model. A grey-level segmentation with small region has been used in order to deal with this constraint. However, it is very likely that some regions will not fulfill this constraint. For problems requiring high accuracy in the segmentation of the motion, a refining process can be performed. The aim of this process is to refine the classification of the pixels without taking into account the initial classification in regions from the given grey-level segmentation. Now, we use the term **outlier** as a pixel that does not support the model, and **inlier** as a pixel that supports the model.

The input of the refining process is the output of our algorithm, i.e. a set of models Υ, each one is made up of a vector of motion parameters χ and a set of regions Φ which support the motion. The refining process consists of:

1. **Find outliers**: For each extracted model M_j, find all the pixels that can be considered as outliers. They are the pixels p_i which their likelihood respect to the model M_j, $L_{M_j}(p_i)$ (see Equation 4.3) is less than a threshold. All the outlier pixels are included in a set, together with the pixels belonging to the region which have been considered outliers at the end of the original algorithm.

2. **Improve parameters**: The motion parameters for the motion models that have new outliers are improved (see subsection 4.4.1).

3. **Find Inliers**: For each outlier, test if it can be included in some of the motion models. A pixel p_i will be included in the model with the greatest likelihood $L_{M_j}(p_i)$, if it is bigger than a threshold and there is a neighbourhood relationship between the pixel p_i and the model M_j. The pixel p_i is neigbour of the model M_j if any pixel into a window of 5x5 centered in p_i belongs to the model M_j.

4. **Improve parameters**: The motion parameters for the motion models that have new inliers are improved (see subsection 4.4.1).

5. **Repeat**: Repeat 1 to 4 while there are changes in the set of pixels.

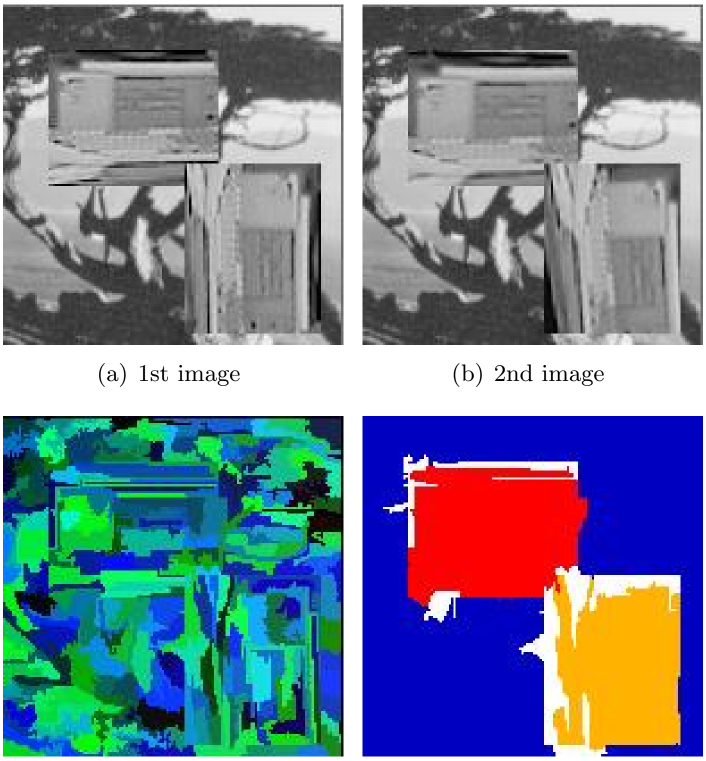
At the end of the refining step the pixels have been classified into the different motion models corresponding to the moving objects in the scene. The pixels that could not be included in any model will be considered as outliers.

4.5 Experimental results

In order to show the performance of the approach presented, two types of experiments have been done. In the first experiment, synthetic sequences have been used, where the results of the motion segmentation and the motion parameters of each model are known. In this synthetic sequence three different motion models can be found. The first is the background, which does not perform motion, i.e. it is static. The second motion model performs a change of scale and the third corresponds to a rotational motion.

In the second experiment real scenes are used, where the final motion segmentation and the motion parameters of each model are unknown. The main motions of the real scene are the background produced by the camera motion, the motion of the car and the motion of the wheels.

Figure 4.13 shows both images of the synthetic sequence, the initial gray segmentation used and the final segmentation obtained, where each final motion model have been labelled with a different RGB color. Figure 4.14 shows both images of the real sequence, the initial gray segmentation used and the final segmentation obtained. White pixels in subfigures 4.13d and 4.14d are the ones that have not been classified in any model. These regions correspond mainly to regions belonging to occluded areas due to the motion and to regions that do not



(a) 1st image

(b) 2nd image

(c) Initial gray segmentation

(d) Final segmentation

Figure 4.13: *Both images of the synthetic sequence and results*

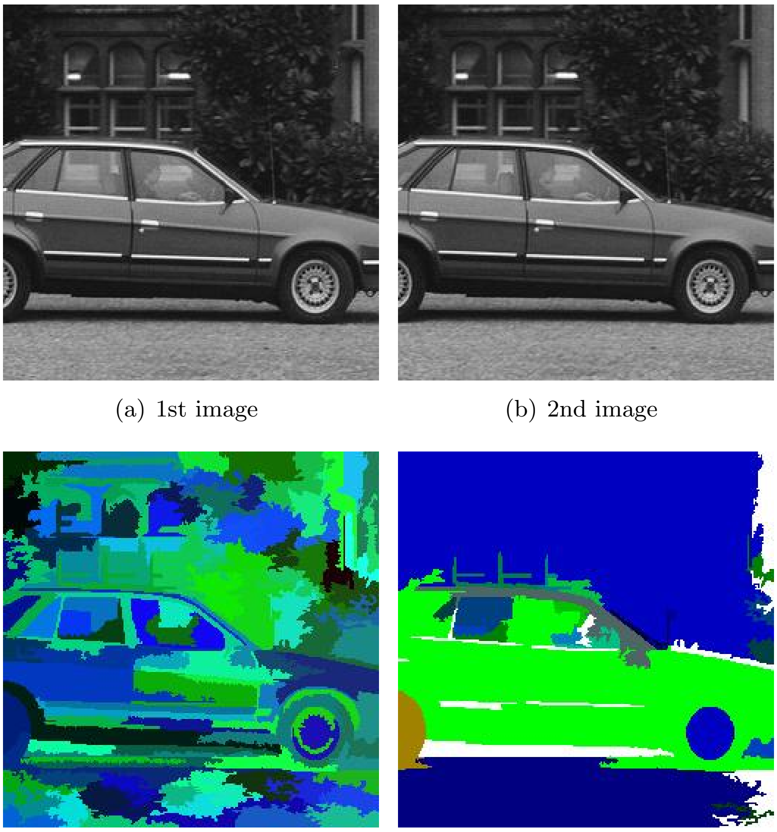
fulfill the requirement of belonging only to a model, i.e. some pixels belong to a model and some other belong to a different model.

Figures 4.15(a) and 4.15(b) show the optic flow for both sequences. They have been computed using the motion parameters of each model in all the pixel belonging to them. They are presented in order to illustrate the motion models estimated.

In order to test the accuracy of the model, two measures P_{WS} and P_{WME} have been calculated. P_{WS} and P_{WME} are expressed as follows:

$$P_{WS} = \frac{N_{ws}}{N} * 100 \tag{4.5}$$

$$P_{MWE} = \frac{N_{wme}}{N} * 100 \tag{4.6}$$



(a) 1st image

(b) 2nd image

(c) Initial gray segmentation

(d) Final segmentation

Figure 4.14: *Both images of the real sequence and results*

where N_{ws} is the number of pixel that have been well-classified with respect to an ideal segmented image, N_{wme} is the number of pixels where the motion have been well estimated and N is the total number of pixels of the input image. That is, P_{WS} and P_{WME} are the percentage of pixels that have been well-classified and the percentage of pixels where the motion have been well estimated, respectively.

For this purpose, the second image of the sequence is compared with a new image generated from the first image of the sequence using the motion parameters of each motion model found. So, P_{WME} is the percentage of pixels where the difference of grey level in both images is less than a threshold, i.e. the percentage of static pixels.

For the synthetic sequence $P_{WS} = 91.5\%$ and $P_{WME} = 99.7\%$. The three motion models have been accurately segmentated and their corresponding motion parameters are also accurately estimated. The main difficulties in the synthetic sequence are the regions that have pixels belonging to more than one model and the regions in occluded areas due to the motion. They have been correctly classified as member of the outliers set.

For the real sequence $P_{WME} = 88.1\%$. The main motions of this sequence have been segmentated, they are the background and the motion of the car. The main difficulties with the real scene are the motion of the wheels, since although our method has detected a rotational motion, it has less magnitude than the real rotation. Nevertheless, interesting results have been obtained in the windows, detecting the motion of the background and the motion of the driver. The outliers are also mainly detected in regions that have pixels belonging to more than one final model and in the regions in occluded areas.

Figure 4.16 shows the results obtained from the two sequences after the refining process. Note that segmentation have been improved in the motion boundaries. Now white pixels are the ones considered as outliers. They are mainly pixels belonging to occluded areas due to the motion and pixels where our algorithm could not estimate the motion due to lack of texture or to the presence of too large motions.

4.6 Conclusions

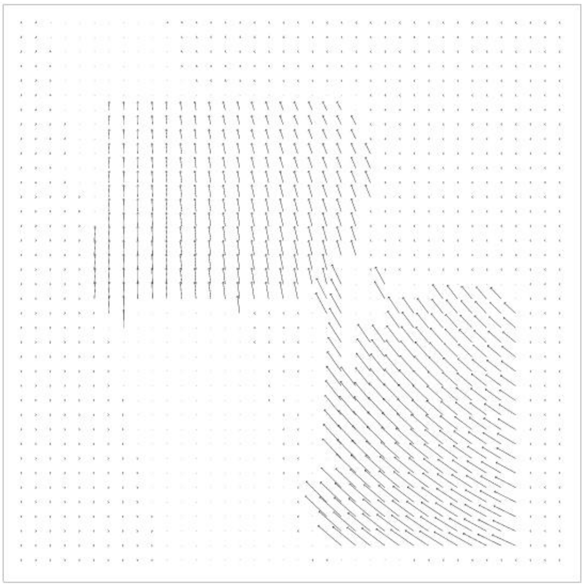
In this chapter, a motion segmentation and estimation algorithm has been presented, which can extract different moving regions of the scene quasi-simultaneously and without a priori information of the number of moving objects. The main properties of our approach are:

- A GLS Motion Estimation algorithm is used, which produces accurate estimation of the motion parameters.

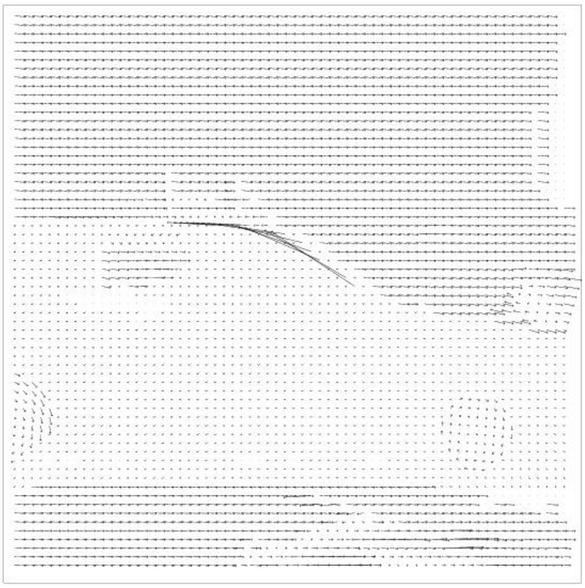
- The classification process which collects inliers, rejects outliers and exchanges regions among models allows to improve motion segmentation.

- It uses regions of pixels instead of pixels as observations and neighbour information, that improves the spatial consistency.

- After motion models have been obtained, a refining process can be used in order to improve segmentation in regions from the initial grey-level segmentation that have pixels belonging to more than one final model.

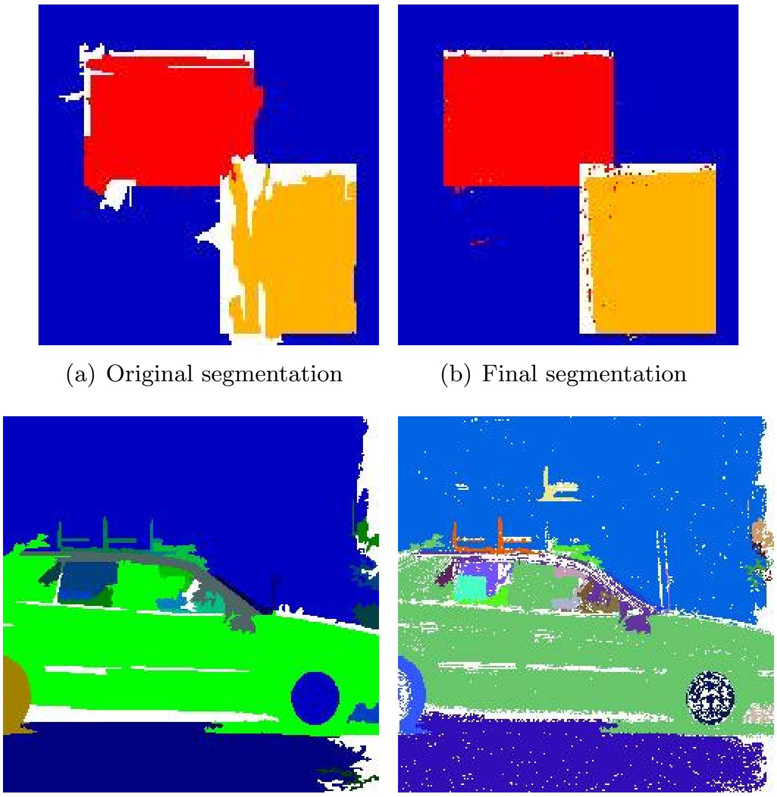


(a) Synthetic



(b) Real

Figure 4.15: *Optic flow computed from results of the synthetic and real sequence*



(a) Original segmentation

(b) Final segmentation

(c) Original segmentation

(d) Final segmentation

Figure 4.16: *Refined segmentation for test sequences*

Chapter 5

Contributions, conclusions and future work

Contents

5.1 Summary of contributions and main conclusions . . . 105
5.2 Future Work 108
5.3 Publications 108

THIS chapter summarizes the main contributions and the conclusions of this work. In addition, future work is also commented. The last part of the chapter mentions the main papers that have been published related to this work.

5.1 Summary of contributions and main conclusions

This thesis has focused on global motion estimation for image registration and motion estimation and segmentation problems. The main contributions of this work can be summarized as follows:

- **Motion estimation**: Regarding the motion estimation problem, we have outlined the problem and the difference between the global an local motion estimation problems. Focusing on global motion estimation for image registration, we have reviewed the literature and studied a number of different techniques. Some of them have been used for comparison purposes.
- **Generalized least squares-based global motion estimation**: We have proposed a GLS mathematical framework to be applied to global motion

estimation problems. In this sense, we have proposed a new GLS-based motion estimation technique that has been successfully applied to image registration and motion segmentation techniques.

One of the key point of the proposed formulation of the motion estimation problem is that ir provides an additional constraint that helps to match the pixels using image gradient information. That is achieved thanks to the use of a weight for each observation, providing high weight values to the observations considered as inliers, i.e. the ones that support the motion model, and low values to the ones considered as outliers.

The main characteristics of the proposed global estimation technique method are summarized as follows:

- It uses a non-linear GLS-motion estimation technique. Therefore, the BCA can directly be used instead of its linearized version, the optical flow equation.
- To avoid falling in a local minimum, it uses a feature-based method (SIFT-based) to obtain good initial motion parameters. Therefore it can deal with large motions.
- The proposed GLS-based motion estimation framework includes an additional constraint, using gradient information, as a way to deal with outliers.
- Similarly to the IRLS technique, this constraint is expressed as a weight to each observation, that varies during the iterative process.

The accuracy of our approach has been tested using challenging real images using affine and projective motion models. Two motion estimator techniques, which use M-Estimators to deal with outliers into an iteratively reweighted least squared-based strategy, have been selected to compare the accuracy of our approach. The results obtained have showed, that the proposed motion estimator can obtain, at least, as accurate results as M-Estimator-based techniques, and even better in most cases.

- **Image registration with large motion**: We have studied the problem of achieving large motion in image registration. We have reviewed some of the most important techniques, most of them related to the extraction of features invariant to rotations, scale changes, small viewpoint changes, etc. To achieve large motion estimation, a two steps algorithm has been proposed, which in the first step uses a feature-based (SIFT-based) technique to obtain a first approximation of the motion parameters. In the second step, the GLS framework is applied to obtain accurate estimates.

- **Image registration under non-uniform illumination**: We have studied the problem of registering two images in the presence of non-uniform illumination changes. A dynamic image model has been used replacing the BCA as objective function, where the illumination factors are function of the localization (x, y) instead of constants, allowing for a more general and accurate image model.

  The use of the dynamic image model instead of the original BCA has provided very satisfactory results. Experiments using challenging real images have been performed to show that, with the combination of both techniques, a global motion estimator can be obtained, which can perform the motion estimation task in an accurate manner while allowing large deformation and illumination changes between images.

- **Motion estimation and segmentation**: Regarding to the motion segmentation problem, we have also reviewed the literature and studied a number of different techniques. A new framework for the motion segmentation and estimation task on sequences of two grey images, without a priori information of the number of moving regions present in the sequence, has been proposed. The proposed algorithm uses temporal information, by using the proposed GLS global motion estimation process and spatial information, by using an iterative region growing algorithm, which classifies regions of pixels into the different motion models present in the sequence. The main properties of our approach are:

  – The proposed GLS Motion Estimation algorithm is used, which produces accurate estimation of the motion parameters.

  – The classification process that collects inliers, rejects outliers and exchanges regions among models, allows to improve motion segmentation.

  – It uses regions of pixels instead of pixels as observations and neighbour information, which improves the spatial consistency.

  – After the motion models present in the image have been obtained, a refining process can be used in order to improve segmentation in regions from the initial grey-level segmentation that have pixels belonging to more than one final model.

5.2 Future Work

Although interesting contributions have been done, much work is still possible, either to improve the proposed algorithms and approaches, or to further explore new directions. A summary of both short- and long-term future work follows:

- **Making the algorithms faster**: Although the algorithms presented at this thesis have been carefully implemented, it is already possible to avoid some not esencial calculations to improve the speed of the processes.

- **Testing the algorithm using other parametric motion models**: In this work, affine and projective motion models have been used. However, other parametric motion models, like the quadratic one, could easily be used in some specific problem where neither projective or affine motion models are adequate.

- **Allowing bigger deformations**: Although the degree of deformation the proposed algorithm can deal has a very important magnitude, future work must focus on researching more sophisticated techniques to allow even bigger deformations. For instance, it could be very interesting to improve the scale change degree and also to allow to register images that have been captured from very different viewpoint changes.

- **Allowing stronger illumination changes**: As in the previous point, it could be very interesting to investigate on obtaining techniques to allow stronger illumination changes between images.

- **Using more temporal information**: The proposed motion segmentation algorithm only uses two consecutive frames. It could be interesting to explore the effect of allowing two use more than to frames, which could probably help to improve the segmentation process.

5.3 Publications

From the work performed i this thesis, some papers have already been published. Figure 5.1 shows the relationship between these publications. The list of publications ordered chronologically is the following:

- **[ICIP 01]** ▶ [Montoliu and Pla, 2001a]

Montoliu R., Pla F. "Multiple Parametric Motion Model Estimation and Segmentation". 2001 International Conference on Image Processing (ICIP'2001), Vol. II, pp. 933-936, ISBN: 0-7803-6725-1, Thessaloniki (Greece), 2001.

- **[CAEPIA 01]** ▶ [Montoliu and Pla, 2001b]

  Montoliu R, Pla F. "Parametric motion model extraction and estimation" IX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2001), Vol. 2, pp. 725-734, Gijón (Spain), ISBN 84-032297-0-9, November 2001.

- **[VIIP 02]** ▶ [Montoliu et al., 2002]

  Montoliu R., Traver V.J., Pla F. "Log-Polar Mapping in Generalized Least-Squares Motion Estimation" 2002 IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP'2002), pp 656-661 ISBN 0-88986-354-0, Málaga (SPAIN), September 2002.

- **[RECPAD 02]** ▶ [Montoliu and Pla, 2002]

  Montoliu R, Pla F. "Quasi-Simultaneous Motion Segmentation and Estimation Using a Generalized Least-Squares Method" 12th Portuguese Conference on Pattern Recognition (RECPAD 2002) Aveiro (Portugal), ISBN 972-789-067-9, June 2002.

- **[VIIP 03]** ▶ [Montoliu and Pla, 2003a]

  Montoliu R., Pla F. "Comparing Brightness Constancy Assumption and Optic Flow Equation in Motion Estimation Algorithms". 3rd International Conference on Visualization, Imaging, and Image Processing. (VIIP'2003), pp 90-95, ISBN 0-88986-382-2. 8-10th September 2003. Benalmádena (SPAIN).

- **[LNCS2905 03]** ▶ [Montoliu and Pla, 2003d]

  Montoliu R., Pla F. "Robust Techniques in Least Squares-Based Motion Estimation Problems". Lecture Notes in Computer Computer Science 2905, Progress in Pattern Recognition, Speech and Image Analysis, A. Sanfeliu and J. Ruiz-Schulcloper (Eds), Springer-Verlag, pp 62-70, ISBN 3-540-20590-X. 2003.

- **[LNCS2652 03]** ▶ [Montoliu and Pla, 2003b]

  Montoliu R., Pla F. "Multiple segmentation of moving objects by quasi-simultaneous parametric motion estimation". Lecture Notes in Computer

Computer Science 2652, Pattern Recognition and Image Analysis, F.J. Perales et al (Eds), Springer-Verlag, pp 572-579, ISBN 3-540-40217-9. 2003.

- **[FAIA 03]** ▶ [Montoliu and Pla, 2003c]

  Montoliu R, Pla F. ”Quasi-Simultaneous Motion Segmentation and Estimation Using an Iterative Region Growing Algorithm” Frontiers in Artificial Intelligence and Applications vol. 100, Artificial Intelligence Research and Development. Isabel Aguiló et al (Eds) IOS Press, pp 189-198. ISBN 1-58603-378-6. 2003.

- **[IJIS 05]** ▶ [Montoliu and Pla, 2005]

  Montoliu R., Pla F. ”An Iterative Region Growing Algorithm for Motion Segmentation and Estimation”. International Journal of Intelligent Systems. Vol 20, Issue 5, pp 577-590. 2004.

- **[LNCS3522 05]** ▶ [Montoliu et al., 2005]

  Montoliu R., Pla F., Klaren A. ”Illumination Intensity, Object Geometry and Highlights Invariance in Multispectral Imaging”. In Lecture Notes in Computer Science 3522. 1, Pages 36-43. 2005.

- **[VISAPP 07]** ▶ [Montoliu and Pla, 2007a]

  Montoliu R., Pla F. ”Accurate Image Registration by Combining feature-based Matching and GLS-based Motion Estimation”. Second International Conference on Computer Vision Theory and Applications. pp 386-389, ISBN 978-972-8865-73-3. 8-11th March. Barcelona, Spain. 2007.

- **[CVIU 07]** ▶ [Montoliu and Pla, 2007b]

  Montoliu R., Pla F., V.J. Traver ”Generalized Least Squares-bases parametric motion estimation”. Submitted to Computer Vision and Image Understanding. 2007

- **[ICIAR 08]** ▶ [Montoliu and Pla, 2008]

  Montoliu R., Pla F., ”Generalized Least Squares-based Parametric Motion Estimation Under Non-uniform Illumination Changes”. International Conference on Image Analysis and Recognition. June 25-27, 2008. Povoa de Barzin, Portugal.

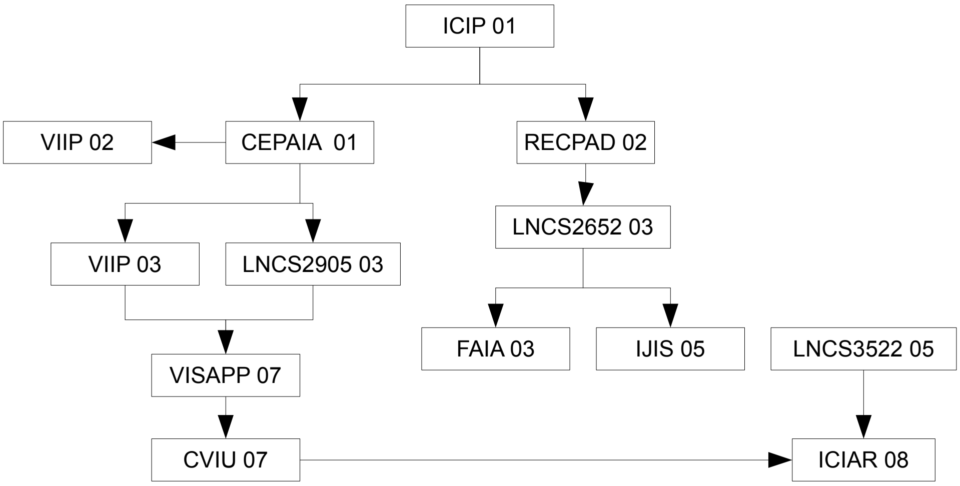


Figure 5.1: *Relation between publications arisen so far from this PhD.*

Appendix A

Test images

THE set of challenging images used in this document can be downloaded from Oxford's Visual Geometry Group web page [1]. They present five types of changes between images in 6 different sets of images: Blur: *Bikes* set, global illumination: *Leuven* set, jpg compression: *Ubc* set zoom+rotation: *Bark* and *Boat* sets and finally viewpoint changes: *Graf* set.

The scale change (*Bark* and *Boat* sets) and blur (*bikes* set) sequences were acquired by varying the camera zoom and focus respectively. The scale changes by about a factor of four. The light changes (*leuven* set) are introduced by varying the camera aperture. The JPEG sequence (*Ubc* set) is generated using a standard *xv* image browser with the image quality parameter varying from 40% to 2%. In *Graf* set the camera varies from a fronto-parallel view to one with significant foreshortening at approximately 60 degrees to the camera. Each image set has 6 different images. The images are showed in Figures A.1, A.2, A.4, A.3, A.5 and A.6.

The 6 previous image sets are the ones that have been used in the experiments presented in Chapter 2. However, additional registration experiments have been performed to show the behavior of the proposed technique. The input images are showed in Figures A.7 (*Tree* image set) and A.8 (*Wall* image set). They can be also downloaded from Oxford's Visual Geometry Group web page. The main deformation of the first image set is due to blur plus a short rotational deformation. The second image set is one of the most difficult, due to the presence of very strong viewpoint.

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

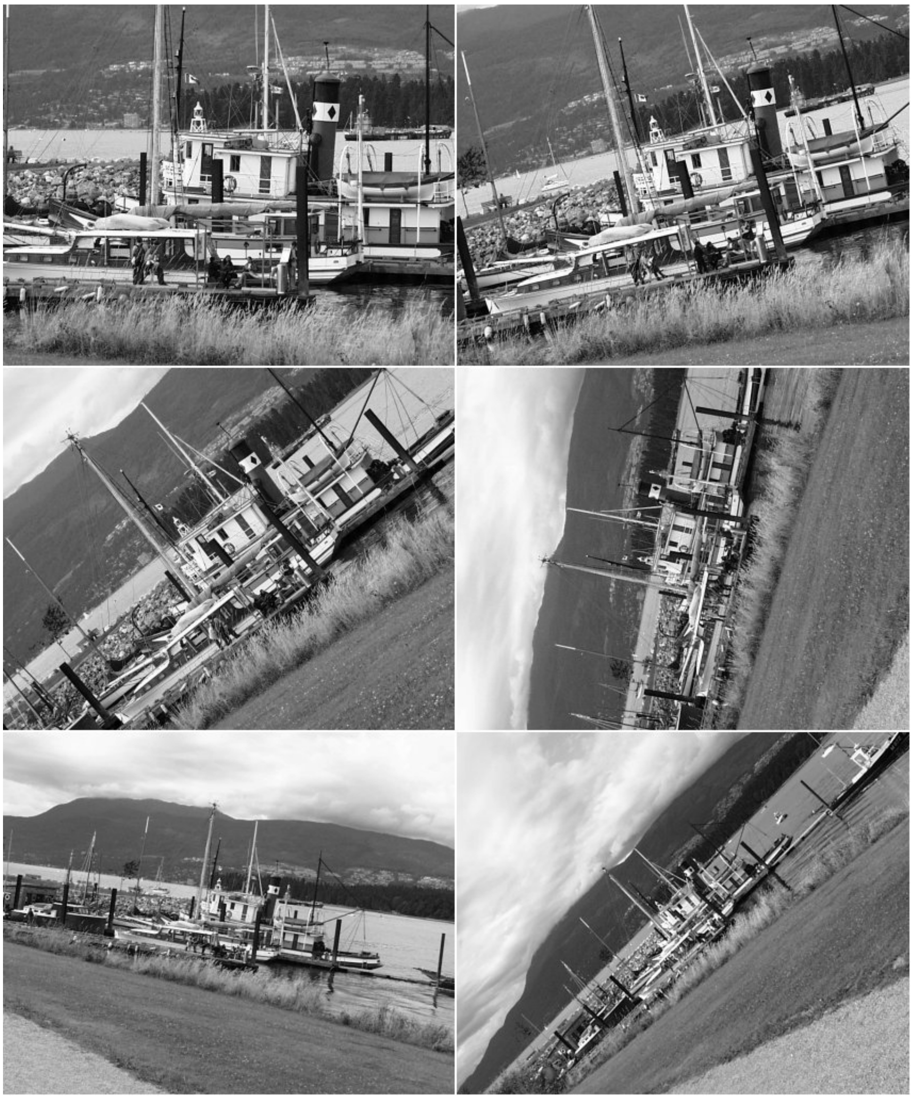


Figure A.1: *Images from* Boat *image set. The changes between images are mainly due to the presence of strong rotations and scale changes.*

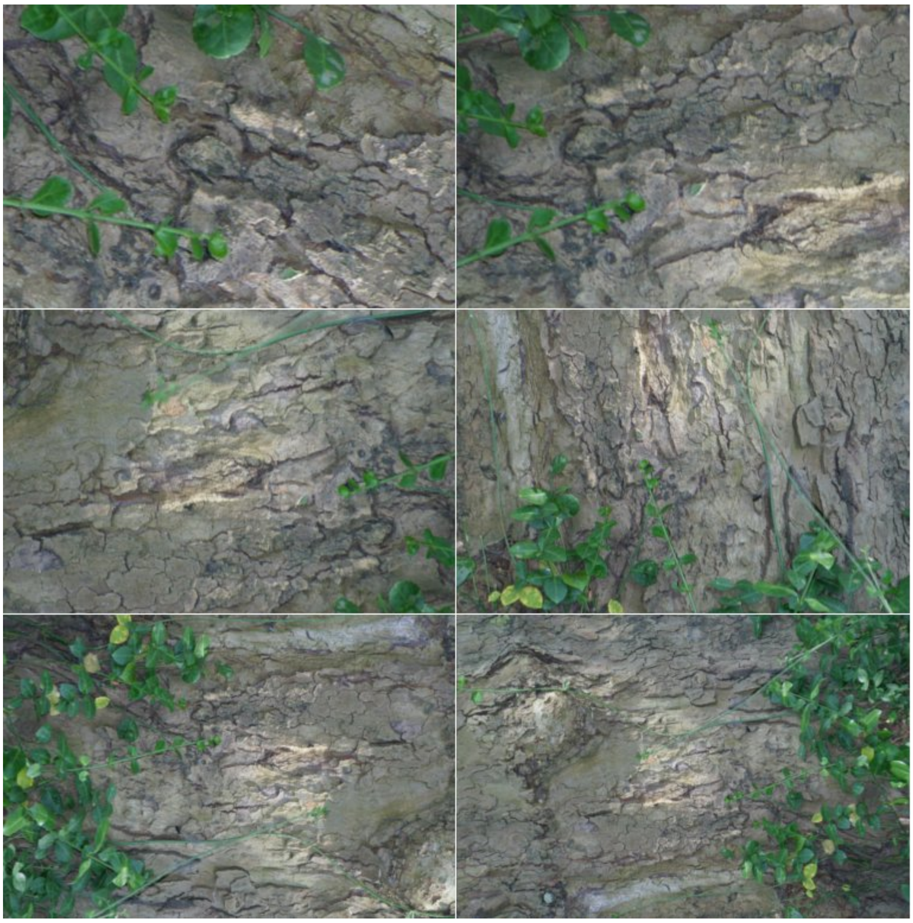


Figure A.2: *Images from* Bark*image set. The changes between images are mainly due to the presence of strong rotations and scale changes.*

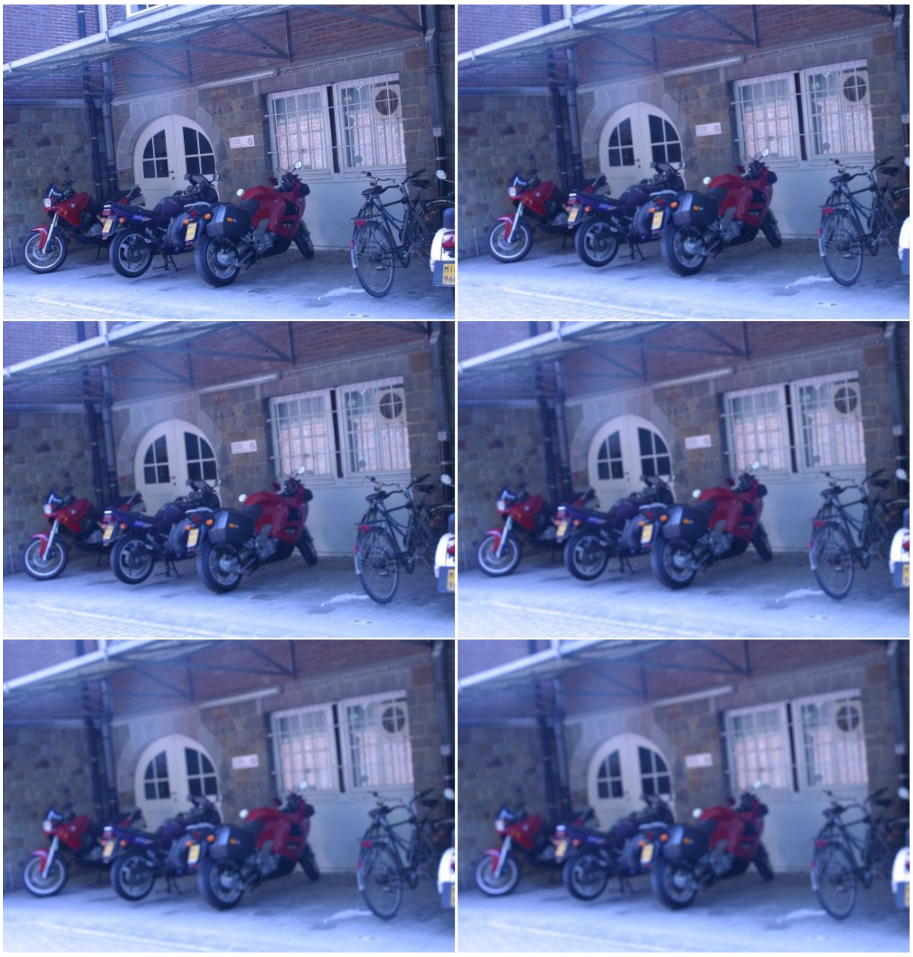


Figure A.3: *Images from* Bikes *image set. The changes between images are mainly due to the presence of blur.*

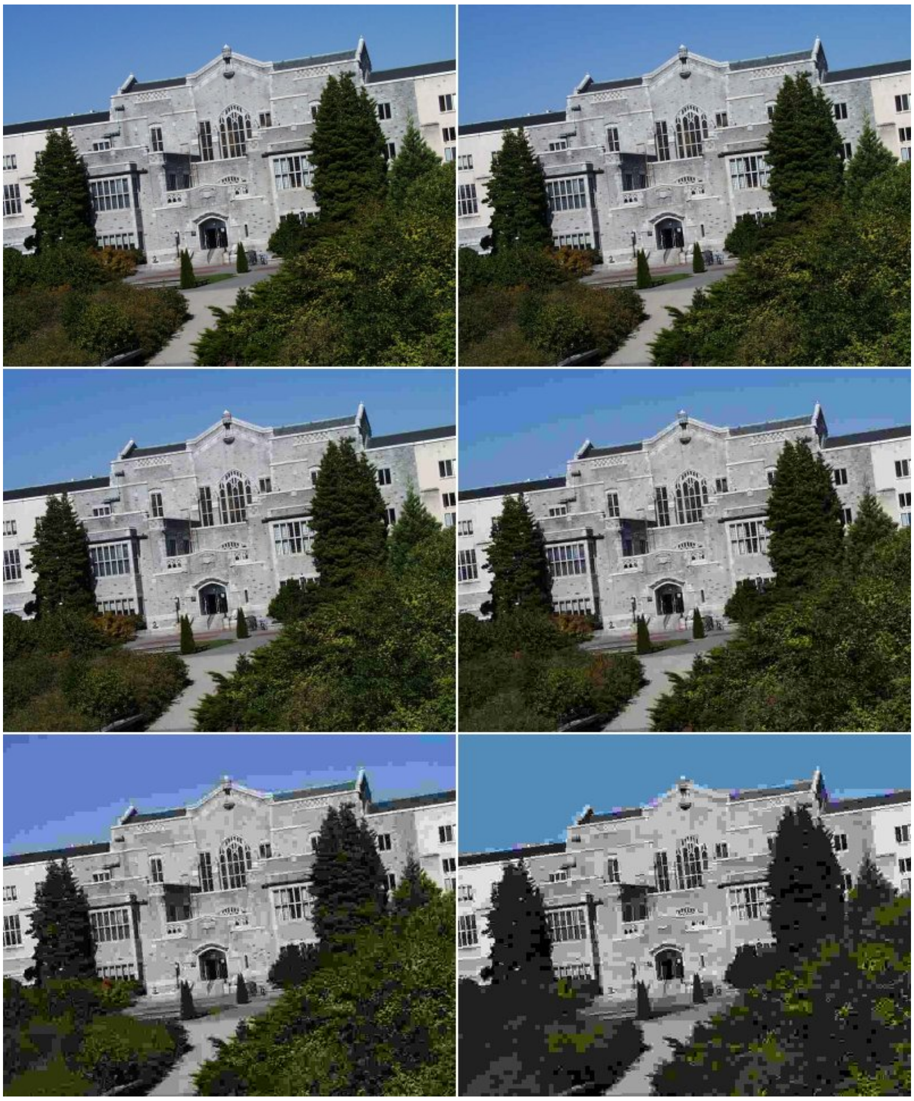


Figure A.4: *Images from* Ubc *image set. The changes between images are mainly due to the use of different compression levels.*

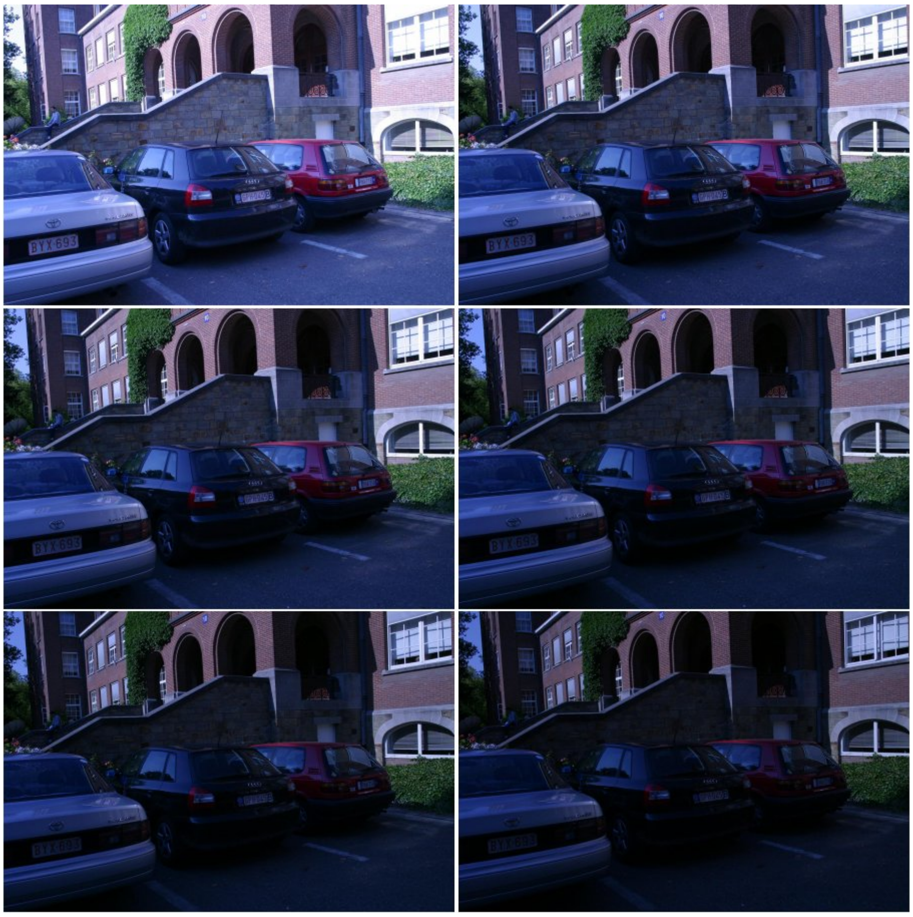


Figure A.5: *Images from* Leuven *image set. The changes between images are mainly due to the presence of global illumination changes.*

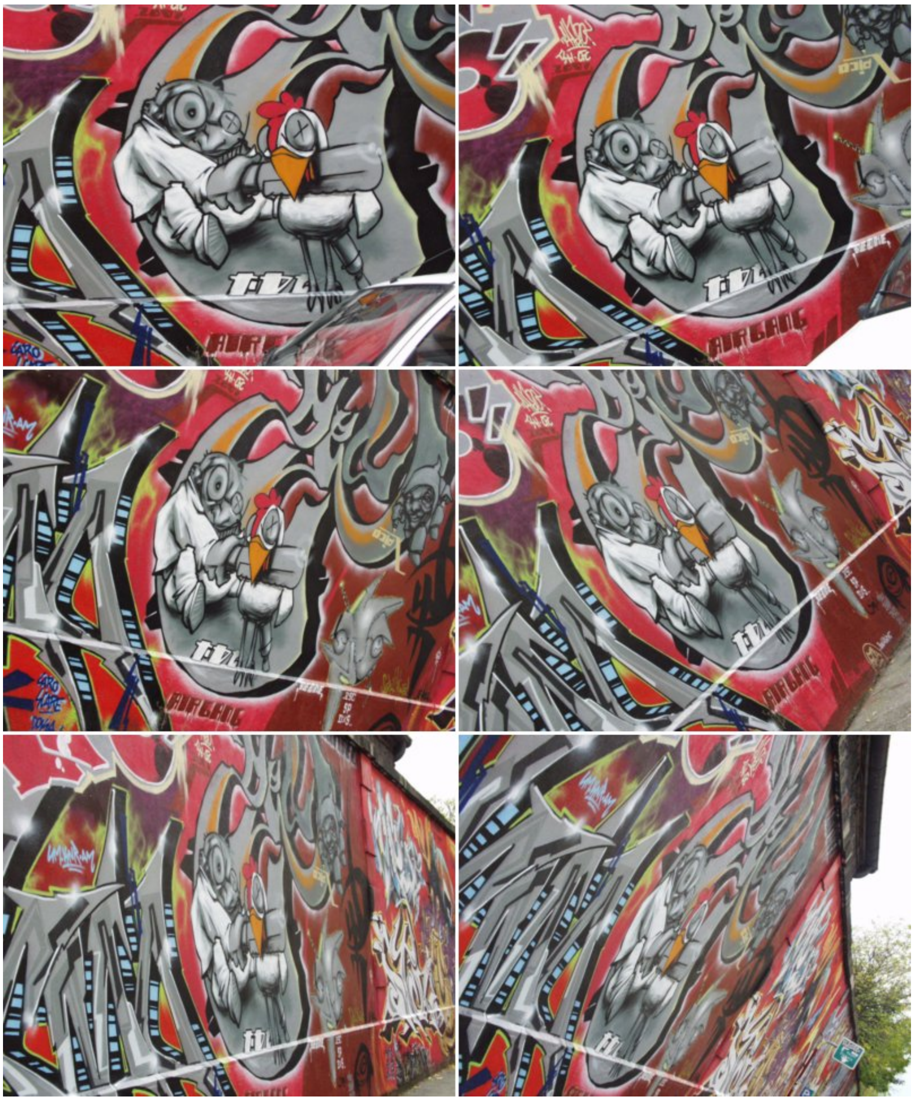


Figure A.6: *Images from* Graf *image set. The changes between images are mainly due to the presence of viewpoint changes.*

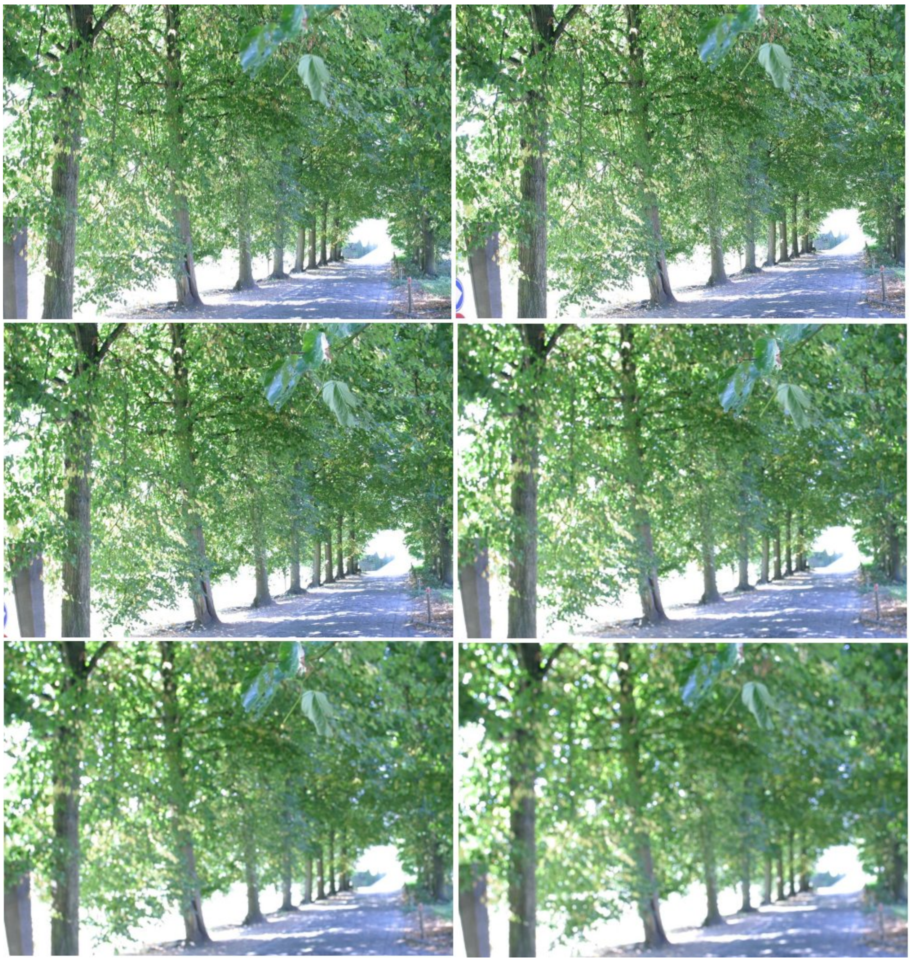


Figure A.7: *Images from* Tree *image set. The changes between images are mainly due to the presence of blur and a short rotation.*

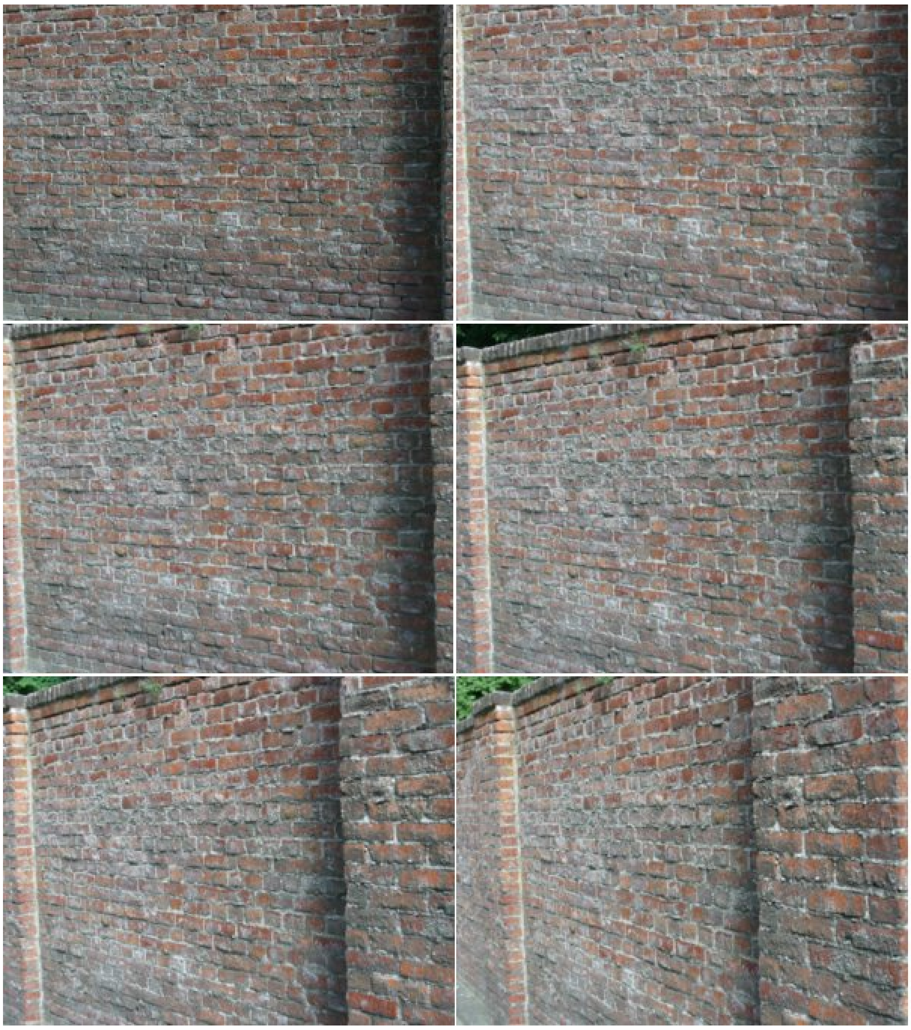


Figure A.8: *Images from* Wall *image set. The changes between images are mainly due to the presence of very strong viewpoint changes.*

Appendix B

Image registration results

This appendix shows several image registration results where the input images have been taken from the image sets showed in Appendix A. Figures B.2 and B.2 show results of *Boat* image set. Figures B.3 and B.4 show results of *Bark* image set. Figures B.5, B.6, B.7 and B.8 show results of *Bikes*, *Ubc*, *Leuven* and *Tree* image set, respectively. Figures B.9, B.10, B.11 and B.12 show results of *Graf* image set. Finally, Figures B.13, B.14 and B.15 show results of *Wall* image sets.

To obtain these results, the affine motion model has been used for the image sets: *Boat*, *Bark*, *Bikes*, *Ubc*, *Leuven* and *Tree*. On the other hand, the projective motion model has been used for the image sets: *Graf* and *Wall*.

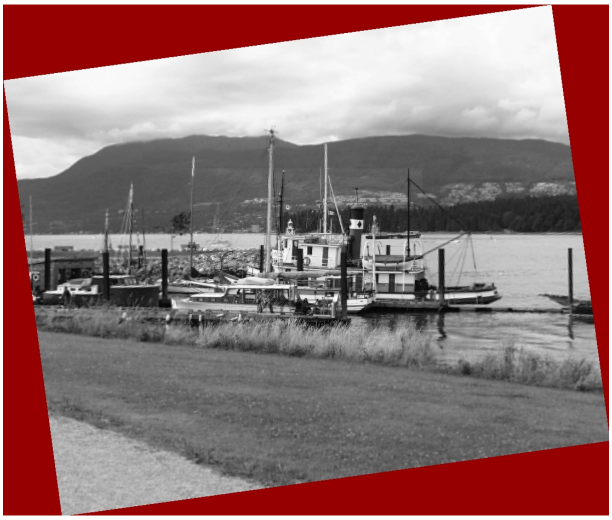


Figure B.1: *Registration results from* Boat *image set. Input images 1st and 5th.*

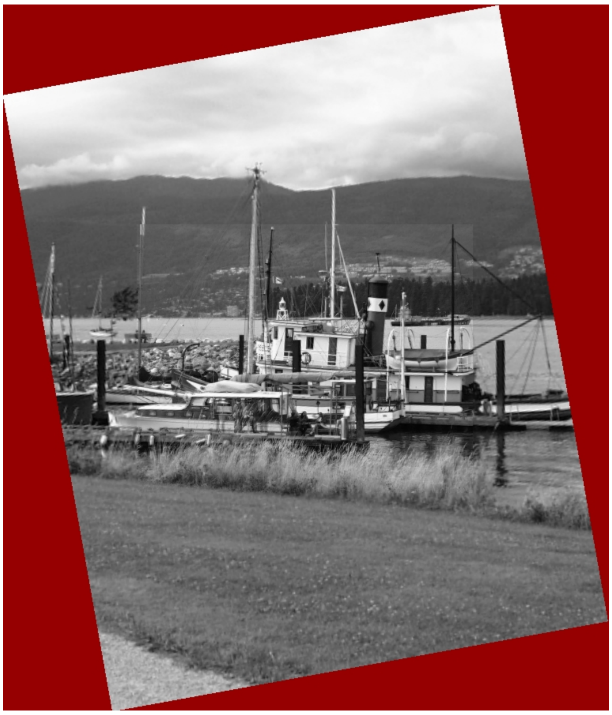


Figure B.2: *Registration results from* Boat *image set. Input images 1st and 4th.*

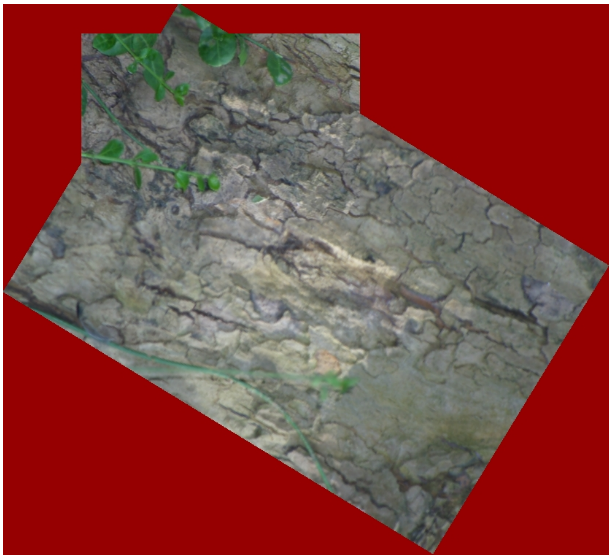


Figure B.3: *Registration results from* Bark *image set. Input images 1st and 3rd.*

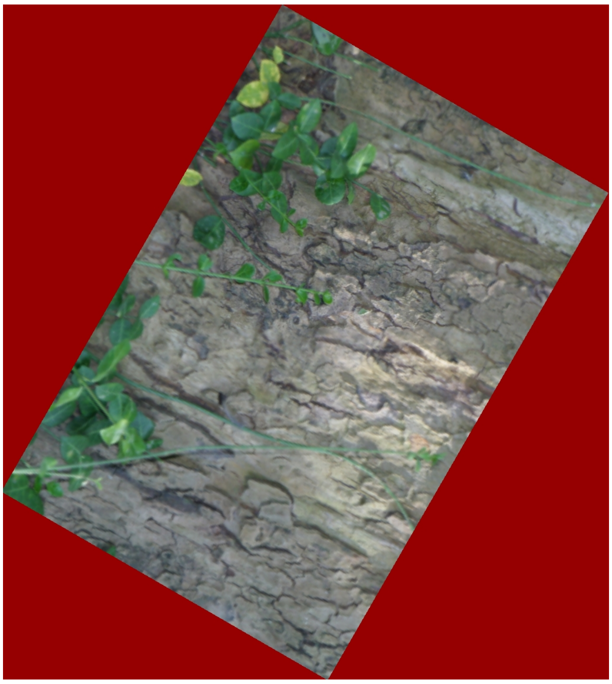


Figure B.4: *Registration results from* Bark *image set. Input images 1st and 4th.*

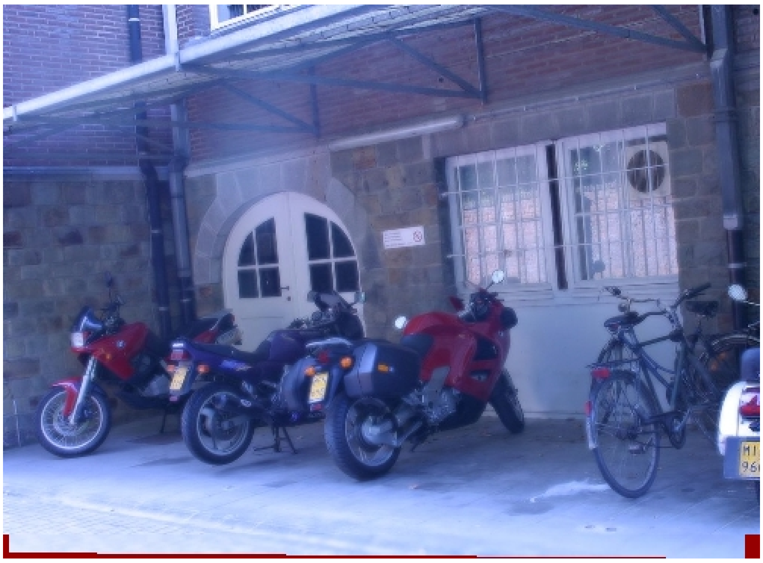


Figure B.5: *Registration results from* Bikes *image set. Input images 1st and 6th.*

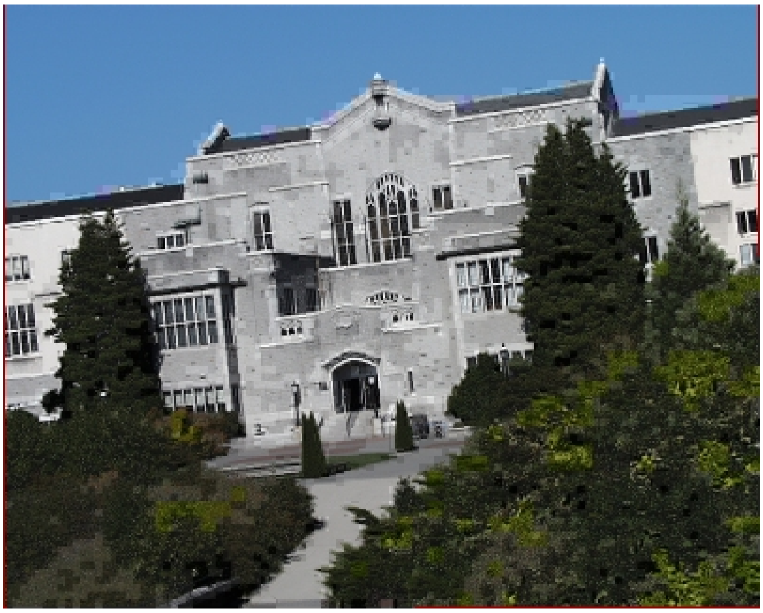


Figure B.6: *Registration results from* Ubc *image set. Input images 1st and 6th.*

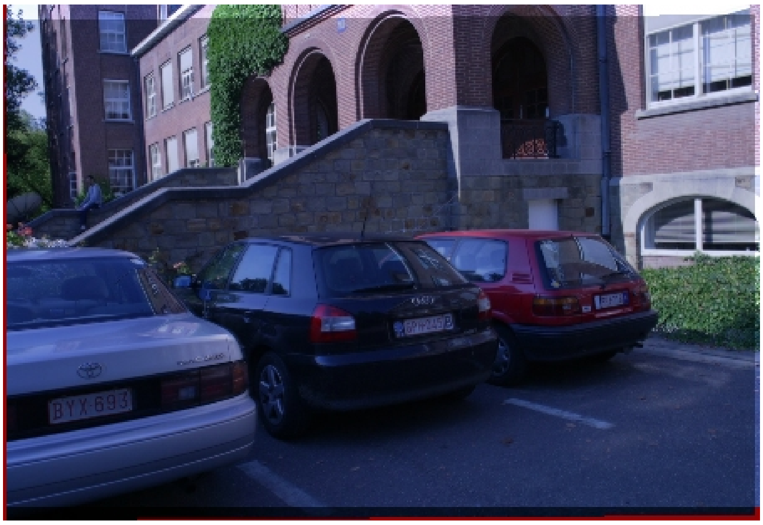


Figure B.7: *Registration results from* Leuven *image set. Input images 1st and 6th.*

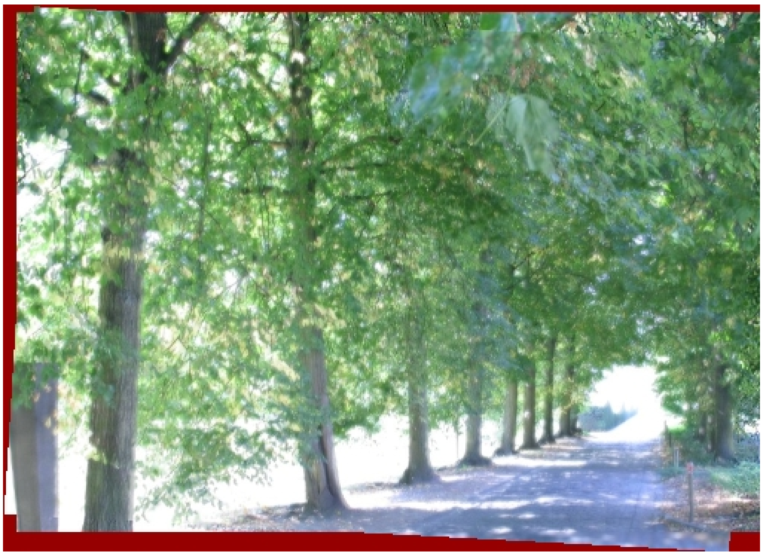


Figure B.8: *Registration results from* Tree *image set. Input images 1st and 6th.*

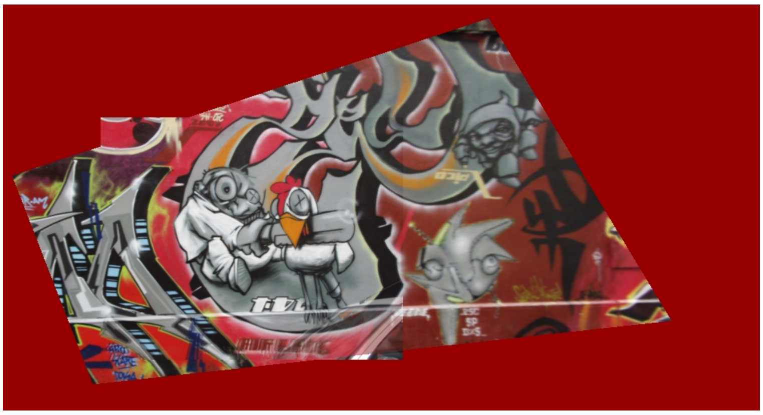


Figure B.9: *Registration results from* Graf *image set. Input images 1st and 3rd.*

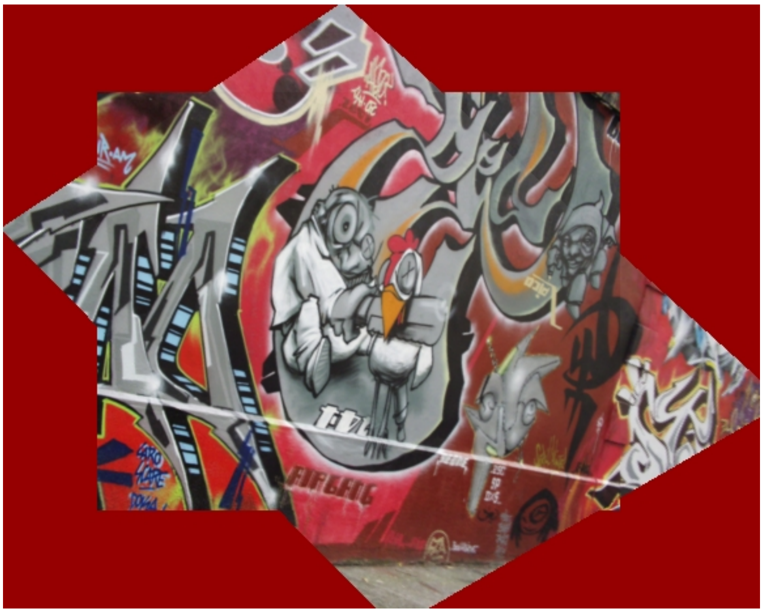


Figure B.10: *Registration results from* Graf *image set. Input images 3rd and 4th.*

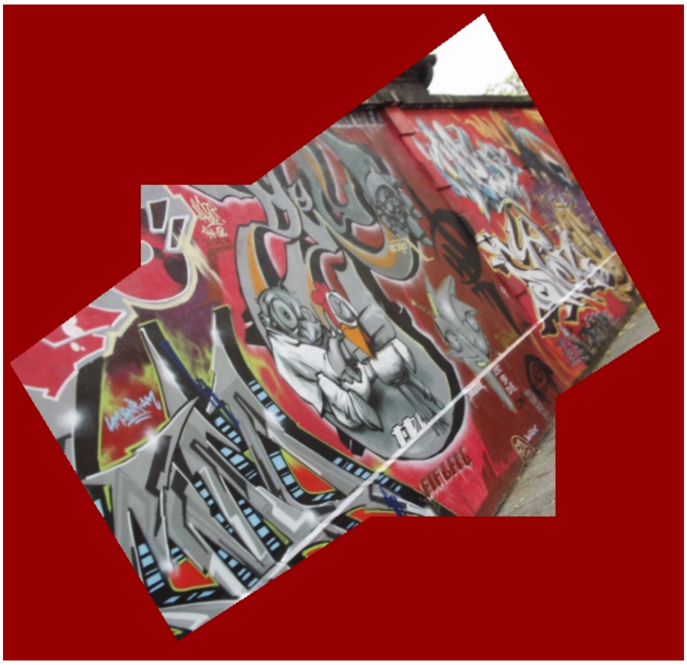


Figure B.11: *Registration results from* GRAF *image set. Input images 4th and 5th.*

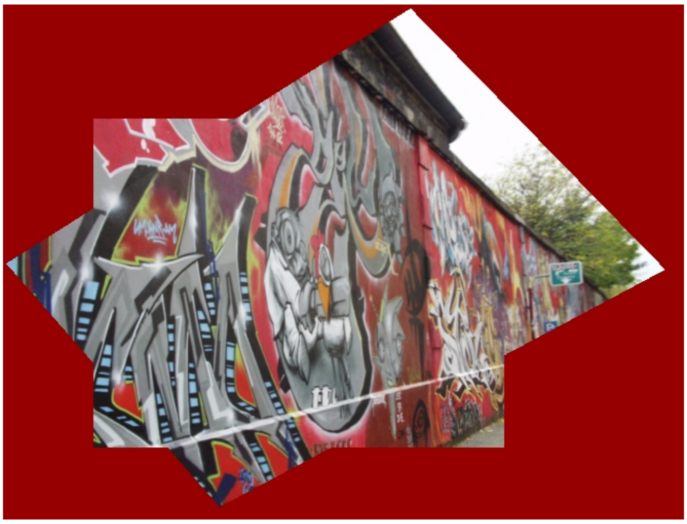


Figure B.12: *Registration results from* Graf *image set. Input images 5th and 6th.*

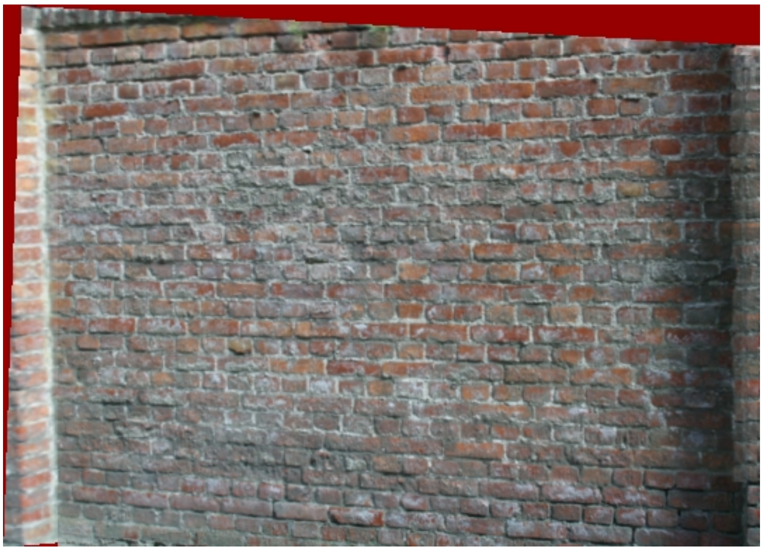


Figure B.13: *Registration results from* Wall *image set. Input images 1st and 3rd.*

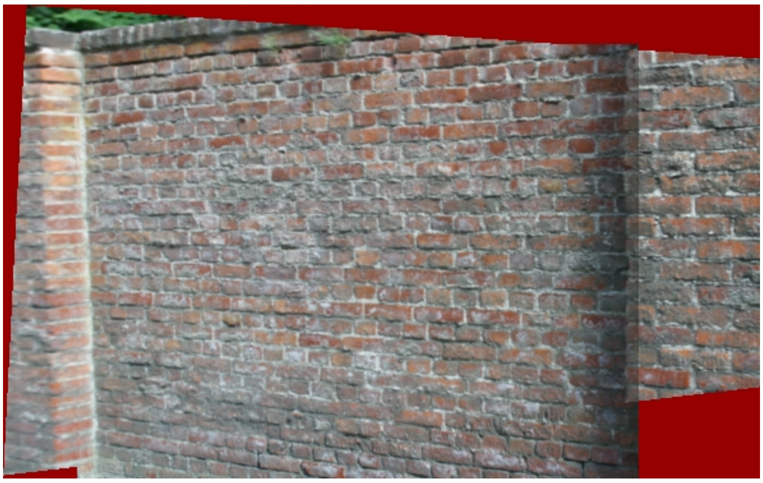


Figure B.14: *Registration results from* Wall *image set. Input images 2on and 5th.*

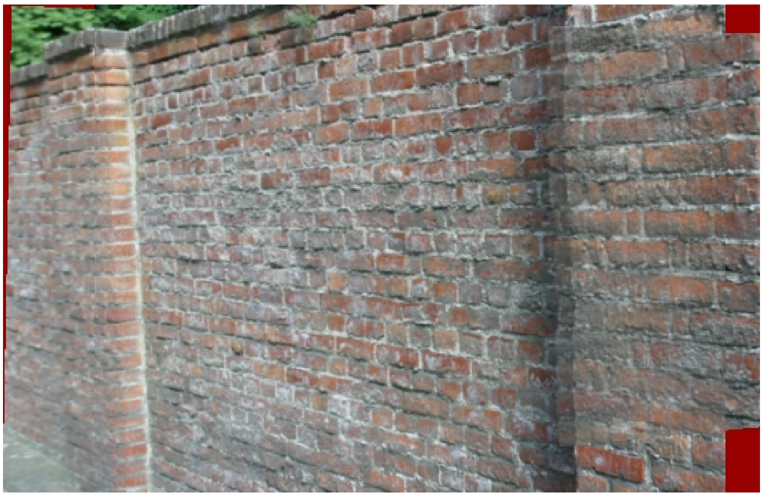


Figure B.15: *Registration results from* Wall *image set. Input images 5th and 6th.*

Bibliography

Ayer, S. and Sawhney, H. S. (1995). Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *IEEE International Conference of Computer Vision*, pages 777–784.

Ayer, S., Schroeterand, P., and Bigun, J. (1994). Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Third European Conference on Computer Vision, ECCV94*, pages 316–327.

Baarda, W. (1968). A testing procedure for use in geodetic networks. *Publications on Geodesy*, 2(5):1–97.

Bad-Hadiashar, A., Gheissari, N., and Suter, D. (2002). Robust model based motion segmentation. In *16th Internation Conference on Pattern Recognition ICPR02, Quebec, Canada*, pages 753–757.

Bad-Hadiashar, A. and Suter, D. (1998). Robust optic flow computation. *International Journal on Computer Vision*, 29(1):59–77.

Baker, S. and Maththews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255.

Baker, S. and Matthews, I. (2002). Lucas-kanade 20 years on: A unifying framework: Part 1. Cmu-ri-tr-02-16, Robotics Institute, Carnegie Mellon University.

Bangham, J. A., Hidalgo, J. R., Harvey, R., and Cawley, G. (1998). The segmentation of images via scale-space trees. In J.N.Carter and N.S.Nixon, editors, *Proceedings of British Machine Vision Conference*, volume 1, pages 33–43, Southampton, UK.

Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *IJCV*, 12(1):43–77.

Bartoli, A. (2006). Groupwise geometric and photometric direct image registration. In *BMVC'06. Proceedings of the Seventeenth British Machine Vision Conference.*

Beauchemin, S. and Barron, J. (1995). The computation of optical-flow. *Surveys*, 27(3):433–467.

Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992a). Hierarchical model-based motion estimation. In *ECCV92*, pages 237–252.

Bergen, J., Burt, P., Hingorani, R., and Peleg, S. (1992b). A three-frame algorithm for estimating two-component image motion. *PAMI*, 14(9):886–896.

Black, M. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Bober, M. and Kittler, J. V. (1993). A hough transform based hierarchical algorithm for motion segmentation and estimation. In Cappellini, V., editor, *4th International workshop on Time-Varying image Processing and Moving Object Recognition*, pages 335–342.

Bober, M. and Kittler, J. V. (1994a). Estimation of complex multimodal motion: An approach based on robust statistics and hough transform. *IVC*, 12(10):661–668.

Bober, M. and Kittler, J. V. (1994b). Robust motion analysis. In *IEEE Conf. on Cpmputer vision and Pattern Recognition.*, pages 947–952.

Britt, H. and Luecke, R. (1973). The estimation of parameters in nonlinear implicit models. *Technometrics*, 15(2):233–247.

Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376.

Brown, M. and Lowe, D. G. (2003). Recognising panoramas. In *In Proceedings of the 9th International Conference on Computer Vision (ICCV2003)*, pages 1218–1225, Nice, France.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In Pajdla, T. and Matas, J., editors, *European Conference on Computer Vision (ECCV)*, volume 3024 of *LNCS*, pages 25–36, Prague, Czech Republic. Springer.

Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337 – 1342.

D'Agostino, E., Maes, F., Vandermeulen, D., and Suetens, P. (2003). A viscous fluid model for multimodal non-rigid image registration using mutual information. *Medical Image Analysis*, 7(4):565–575.

Danuser, G. and Stricker, M. (1998). Parametric model-fitting: From inlier characterization to outlier detection. *PAMI*, 20(3):263–280.

de Castro, E. and Morandi, C. (1987). Registration of translated and rotated images using finite fourier transforms. *PAMI*, 9(5):700–703.

Dufournaud, Y., Schmid, C., and Horaud, R. (2004). Image matching with scale adjustment. *Computer Vision and Image Understanding*, 93(2):175–194.

Ferrier, N., Rowe, S., and Blake, A. (1994). Real-time traffic monitoring. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 81–88.

Finlayson, G. D., Hordley, S. D., and Drew, M. S. (2002). Removing shadows from images. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 823–836, London, UK. Springer-Verlag.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Geusebroek, J. M., van den Boomgaard, R., Smeulders, A. W. M., and Geerts, H. (2001). Color invariance. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(12):1338–1350.

Gevers, T. and Smeulders, A. W. M. (1999). Colour based object recognition. *Pattern Recognition*, 32:453–464.

Grecos, C., Saparon, A., and Chouliaras, V. (2004). Three novel low complexity scanning orders for mpeg-2 full search motion estimation. *Real Time Imaging*, 10:53–56.

Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. (1986). *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons.

Haritaoglu, I., Harwood, D., and Davis, L. (2000). W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809 – 830.

Horn, B. and Schunk, B. (1981). Determining optical flow. *Artificial Intelligence*, 17. (1981).

Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334 – 352.

Huber, P. (1981). *Robust Statistics.* John Wiley and sons.

Illingworth, J. and Kittler, J. (1988). A survey of the hough transform. *CVGIP*, 44(1):87–116.

Irani, M., Rousso, B., and Peleg, S. (1994). Computing occluding and transparent motion. *IJVC*, 12(1):5–16.

J., B., J.M., S., and F., P. (2001). Motion-based segmentation and region tracking in image sequences. *Pattern Recognition*, 34:661–670.

Ji, X., Wei, Z., and Feng, Y. (2006). Effective vehicle detection technique for traffic surveillance systems. *Journal of Visual Communication and Image Representation*, 17(3):647–658.

Kaneko, S., Murase, I., and Igarashi, S. (2002). Robust image registration by increment sign correlation. *Pattern Recognition*, 35(10):2223–2234.

Kaneko, S., Satoh, Y., and Igarashi, S. (2003). Using selective correlation coefficient for robust image registration. *Pattern Recognition*, 36(5):1165–1173.

Kang, H., Lee, C. W., and Jung, K. (2004). Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, 25(15):1701–1714.

Kastrinaki, V., Zervakis, M., and Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4):359–381.

Keller, Y. and Averbuch, A. (2004). Fast motion estimation using bi-directional gradient methods. *IEEE Transactions on Image Processing*, 13(8):1042–1054.

Keller, Y. and Averbuch, A. (2008). Global parametric image alignment via high-order approximation. *Comput. Vis. Image Underst.*, 109(3):244–259.

Kim, J. B. and Kim, H. J. (2003). Effient region-based motion segmentation for a video monitoring system. *Pattern Recognition Letters*, 24:113–128.

Kim, Y., Martinez, A. M., and Kak, A. C. (2004). Robust motion estimation under vaying illumination. *Image and Vision Computing*, 23(4):365–375.

Kubrick, S. and Clarke, A. C. (1968). 2001: A space odyssey. Film script.

Lai, S.-H. and Fang, M. (1999). Robust and efficient image alignment with spatially varying illumination models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 02, page 2167.

Lee, K.-M., Meer, P., and Park, R.-H. (1998). Robust adaptavice segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):200–205.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679.

Lucceche, L., G.M., C., and M., R. (1997). A phase correlation technique for estimation planar rotations. In Cappellini, V., editor, *Time-Varying Image Processing and Moving Object Recognition, 4*, pages 244–249.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10). (2005).

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2). (2005).

Miller, J. V. and Stewart, C. V. (1996). Muse: Robust surface fitting using unbiased scale estimates. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Montoliu, R. and Pla, F. (2001a). Multiple parametric motion model estimation and segmentation. In *International Conference on Image Processing (ICIP'2001) Thessaloniki (Greece)*, volume II, pages 933–936.

Montoliu, R. and Pla, F. (2001b). Parametric motion model extration and estimation. In *Proccedings of the IX Spanish Symposium on Artificial Inteligence*, pages 725–734.

Montoliu, R. and Pla, F. (2002). Quasi-simultaneous motion segmentation and estimation using a generalized least-squares method. In *Proccedings of 12th Portuguese Conference on Pattern Recognition (RECPAD 2002)*.

Montoliu, R. and Pla, F. (2003a). Comparing brightness constancy assumption and optic flow equation in motion estimation algorithms. In *Proccedings of 2003 IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP'2003)*, pages 90–95.

Montoliu, R. and Pla, F. (2003b). *Multiple segmentation of moving objects by quasi-simultaneous parametric motion estimation*, pages 572–579. Lecture Notes in Computer Science 2652, Pattern Recognition and Image Analysis. springer Verlag.

Montoliu, R. and Pla, F. (2003c). *Quasi-Simultaneous Motion Segmentation and Estimation Using an Iterative Region Growing Algorithm*. Frontiers in Artificial Intelligence and Applications vol 100. IOS Press.

Montoliu, R. and Pla, F. (2003d). *Robust Techniques in Least Squares-Based Motion Estimation Problems*, pages 62–70. Lecture Notes in Computer Computer Science 2905, Progress in Pattern Recognition, Speech and Image Analysis.

Montoliu, R. and Pla, F. (2005). An iterative region growing algorithm for motion segmentation and estimation. *International Journal of Intelligent Systems*, 20(5):577–590.

Montoliu, R. and Pla, F. (2007a). Accurate image registration by combining feature-based matching and gls-based motion estimation. In *Second International Conference on Computer Vision Theory and Applications. 8-11th March. Barcelona, Spain*, pages 386–389.

Montoliu, R. and Pla, F. (2007b). Generalized least squares-based parametric motion estimation. Technical Report 1/10/2007, University Jaume I.

Montoliu, R. and Pla, F. (2008). *Generalized Least Squares-based Parametric Motion Estimation Under Non-uniform Illumination Changes*, pages 660–669. Lecture Notes in Computer Science 5112.

Montoliu, R., Pla, F., and Klaren, A. (2005). *Illumination Intensity, Object Geometry and Highlights Invariance in Multispectral Imaging*, pages 36–43. Lecture Notes in Computer Science 3522.

Montoliu, R., Traver, V., and Pla, F. (2002). Log-polar mapping in generalized least-squares motion estimation. In *Proccedings of 2002 IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP'2002)*, pages 656–661.

Negahdaripour, S. (1998). Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979.

Nir, T., Bruckstein, A. M., and Kimmel, R. (2008). Over-parameterized variational optical flow. *International Journal on Computer Vision*, 76(2):205–216.

Odobez, J. M. and Bouthemy, P. (1995). Robust multiresolution estimation of parametric motion models. *Int. J. Visual Communication and Image Representation*, 6(4):348–365.

Odone, F., Fusiello, A., and Trucco, E. (2000). Robust motion segmentation for content-based video coding. In *RIAO 2000 6th Conference on Content-Based Multimedia Information Access*, pages 594–601.

Pearson, J., Hines, D., Golosman, J., and Kuglin, C. (1977). Video-rate image correlation processor. *SPIE, Applications of Digital Image Processing*, 119:197–205.

Periaswamy, S., Weaver, J. B., Healy, D. M., Rockmore, D. N., Kostelec, P. J., and Farid, H. (2000). Differential affine motion estimation for medical image registration. In *SPIE's 45th annual Meeting, The international Symposium on Optical Science and Technology.*

Pizarro, D. and Bartoli, A. (2007). Shadow resistant direct image registration. In *SCIA'07 - Proceedings of the Fifteenth Scandinavian Conference on Image Analysis.*

P.J., R. and A.M., L. (1987). *Robust Regression and Outliers Detection.* John Wiley and Sons.

Pla, F. and Bober, M. (1997). Estimating translation/deformation motion through phase correlation. In *9th. International Conference on Image Analysis and Processing*, pages 653–660. Lecture Notes in Computer Science, A. del Bimbo (Ed.), Springer-Verlag.

Pluim, J. P., Maintz, J. A., and Viergever, M. A. (2003). Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004.

Prati, A., Mikic, I., Trivedi, M., and Cucchiara, R. (2003). Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918 – 923.

Rad, R. and Jamzad, M. (2005). Real time classification and tracking of multiple vehicles in highways. *Pattern Recognition Letters*, 26(10):1597–1607.

Radke, R., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294 – 307.

Ren, L., Shakhnarovich, G., Hodgins, J. K., Pfister, H., and Viola, P. (2005). Learning silhouette features for control of human motion. *ACM Trans. Graph.*, 24(4):1303–1331.

Ricardo A. Maronna, Douglas R. Martin, V. J. Y. (2006). *Robust Statistics: Theory and Methods*. John Wiley and Sons.

Rogelj, P., Kova, S., and Gee, J. C. (2003). Point similarity measures for non-rigid registration of multi-modal data. *Computer Vision and Image Understanding*, (92):112–140.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880.

S. Baker, R. Gross, I. M. and Ishikawa, T. (2003). Lucas-kanade 20 years on: A unifying framework: Part 2. Cmu-ri-tr-03-01, Robotics Institute, Carnegie Mellon University.

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

Serkis, A. (2003). *Gollum: A Behind the Scenes Guide of the Making of Gollum (The Lord of the Rings)*. Houghton Mifflin Co.

Setchell, C. (1997). Applications of computer vision to road-traffic monitoring. Technical Report CS-EXT-1997-118.

Shafer, S. (1984). Using color to separate reflection components. *Journal of the Optical Society of America*, 1:1248–+.

Szeliski, R. (2004). Image alignment and stitching: A tutorial. Technical Report MSR-TR-2004-92, Microsoft Research.

Szeliski, R. and Coughlan, J. (1997). Spline-based image registration. *International Journal of Computer Vision*, 22(3):199–218.

Tai, J.-C., Tseng, S.-T., Lin, C.-P., and Song, K.-T. (2004). Real-time image tracking for automatic traffic monitoring and enforcement applications. *Image and Vision Computing*, 22(6):485–501.

Torr, P. and Zisserman, A. (1997). Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(591–605).

Torr, P. H. S. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156.

Tseng, S.-Y. (2004). Motion estimation using a frame-based adaptative thresholding approach. *Real Time Imaging*, 10:1–7.

Tu, J., Tao, H., and Huang, T. (2007). Face as mouse through visual face tracking. *Computer Vision and Image Understanding. Special Issue on Vision for Human-Computer Interaction,*, 108(1-2):35–40.

Tukey, J. W. (1977). *Exploratory Data Analysis.* Addison-Wesley, Reading, MA.

Wang, L., Hu, W., and Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error measurement to structural similarity. *IEEE Transaction on Image Processing*, 13(4).

Zhang, Z. (1997). Parameter-estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76.

Zitova, B. and Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21:997–1000.