Adaptieve technieken met polynomiale modellen voor segmentatie, approximatie en analyse van gezichten in videosequenties

Adaptive Techniques with Polynomial Models for Segmentation, Approximation and Analysis of Faces in Video Sequences

Francis Deboeverie

Promotoren: prof. dr. ir. W. Philips, prof. dr. ir. P. Veelaert
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2013 - 2014

UNIVERSITEIT
GENT

ISBN 978-90-8578-707-5
NUR 958
Wettelijk depot: D/2014/10.500/53

Adaptieve technieken met polynomiale modellen voor segmentatie, approximatie en analyse van gezichten in videosequenties

Adaptive Techniques with Polynomial Models for Segmentation, Approximation and Analysis of Faces in Video Sequences

Francis Deboeverie

Members of the jury

prof. dr. ir. Jean Martinet (University of Lille)
prof. dr. Rudi Penne (University of Antwerp)
dr. ir. Chris Poppe (TomTom Ghent)
prof. dr. ir. Peter Lambert (Ghent University, secretary)
prof. dr. ir. Patrick De Baets(Ghent University, chairman)
prof. dr. ir. Peter Veelaert (Ghent University, promoter)
prof. dr. ir. Wilfried Philips (Ghent University, promoter)

Affiliations

Research Group for Image Processing and Interpretation (IPI)
Department of Telecommunications and Information Processing (TELIN)

Research Group Vision Systems (VIS)
Department of Industrial Technology and Construction (IT&C)

Faculty of Engineering and Architecture (FEA)
Ghent University (UGent)

iMinds

Acknowledgements

This work is made possible with the help and support of many people.
I would like to thank my supervisors prof. Wilfried Philips and prof. Peter Veelaert for their guidance, support and trust during the past years.
I would like to thank the members of my PhD jury, prof. Jean Martinet, prof. Rudi Penne, dr. Chris Poppe, prof. Peter Lambert, prof. Patrick De Baets, prof. Peter Veelaert and prof. Wilfried Philips, for being in my thesis committee and for giving various comments and suggestions.
I would like to acknowledge all my colleagues at the department of Industrial Technology and Construction and the department of Telecommunications and Information Processing at Ghent University.
I would also like to thank all the persons with whom I collaborated during this PhD.
Last but not least, I would like to thank my wife, Rachida Vanslembrouck, and my parents, Miriam Baele and Ivan Deboeverie, for their support and patience during the work on this PhD.

Ghent, April 2014
Francis Deboeverie

Summary

In today's society, the use of image and video data has become very widespread, due to the increasing availability of high-quality imaging devices, such as digital still image cameras. People use image and video data to share their daily life experiences with other people, for instance on social media. Likewise, immediate remote communication has become very popular. For instance in automated video conferencing, intelligent cameras employ face analysis systems with on-board face databases. In this application, a compact face representation will reduce processing and storage cost. Ideally for video analysis, such a representation would be based on a perfect segmentation of a face into all its major parts, i.e., forehead, nose, eyes, lips, etc. For example, it is important that lip motion follows speech very closely with almost no time delay, which is much easier to accomplish when the image representation already contains the lips as separate items. Furthermore, encoding based on smart face segmentation will facilitate additional tasks such as recognizing faces, facial expressions, and gaze direction.

This PhD research presents a solution for face segmentation, approximation and analysis in visual communication applications by means of computer vision.

- **Face segmentation** is the division of face images into physically meaningful parts, such as the forehead, the cheeks, the lips, the eyebrows, etc. In this work, we propose to group contour pixels and image pixels (grey values, image intensities) in images of faces with region growing and polynomial fitting. Region growing is the process of examining neighbouring pixels of initial seed pixels and determining whether the neighboring pixels should be added to the region. Polynomial fitting fits geometric primitives to pixels. A geometric primitive is a polynomial function describing the geometry of an edge or the variation of grey values in a region. Thus, the problem we study is that of finding a region of maximal size in which grey values can be well approximated by a polynomial function and where contour pixels can be well approximated by polynomials as well. Face segmentation in this work considers the grouping of facial contour pixels into contour segments, as well as the grouping of facial intensities into surface segments.

  To find segments, we propose an adaptive region growing algorithm based on constructive polynomial fitting. This primitive extraction algorithm finds subsets of pixels that lie on a geometric primitive or close to it. How well a subset corresponds to a primitive is quantified by an L_∞ fitting cost (approximation error). An original contribution is the introduction of adaptive

thresholding for region growing, which allows a variable polynomial degree and a variable fitting cost, depending on the local properties of the pixels. The L_∞ fitting cost is the maximum deviation between the pixels and the polynomial function. The novelty is that region growing detects outliers, distinguishes between strong and smooth discontinuities and considers the curvature, such as convexity or concavity. An outlier is a pixel differing from its neighbours due to noise or small object speckles. The region growing investigates the local variation of pixels in a segment to identify outliers, while the global variation of pixels in a segment is investigated to adapt the degree of the polynomial function. The terms local and global refer to a part of the segment and the entire segment, respectively. The combination of both is possible because we employ constructive fitting: the global fitting cost is calculated from local fitting costs. In this work, we demonstrate how local and global fitting costs can interact in an adaptive method.

Whereas in face segmentation the focus of the research is on grouping the pixels, face approximation focuses on producing accurate approximations of the pixels.

- **Face approximation** is the estimation and the reconstruction of contour pixels and grey values of face images to a desired degree of accuracy. In this work, we propose to approximate faces with maps defined by polynomial functions (geometric low-level features, geometric primitives) of low-degree (e.g., 0, 1 or 2). One difficulty we study is that of finding the parameters of the best fit.

  The contour model represents contour segments as polynomial curves, which are either straight, convex or concave, in a Curve Edge Map (CEM). The surface model represents surface segments as polynomial surfaces, which are either flat, planar, convex, concave or saddle surfaces, in a Surface Intensity Map (SIM). Both models are simple, natural, useful and elegant representations for objects in images, in particular for face images. The low-degree polynomial functions in these models provide good approximation of meaningful facial features, while preserving all the necessary details of the face in the reconstructed image.

  The contour and surface models allow parametrizing a face by the coefficients of the polynomial curves and surfaces and the coordinates of the endpoints of the curves in a few hundred bytes. Transmitting these face parameters over the network is very efficient and the method preserves all the necessary details of the face in the reconstructed image. Furthermore, these face descriptors are suitable for automated face analysis.

- **Face analysis** includes recognition and tracking of faces and facial features. Face recognition identifies a person in a digital image or a video frame. Face tracking follows the movements of a person's head in a video. Recognition and tracking of faces and facial features leads to the detection of specific

human face behaviour, such as speaking, gaze direction, head movements (e.g. nodding) or facial expressions (e.g. happy/sad/angry/surprised).

In order to compare faces in two different or consecutive images, we present a technique to find geometric feature correspondence pairs. The goal of a correspondence finding or matching algorithm is to indicate for a point (feature) in one image which is the corresponding point (feature) in a second image, where both image points must show the same 3-D world point.

In the contour model, we find correspondences by a technique which matches polynomial curves, based on shape, relative position and intensity. We propose a dissimilarity function for local curve matching as well as a similarity function for global curve matching. The difference lies in the application: local matching is primarily used in tracking applications, while global matching focuses on recognition applications.

In the surface model, we perform curvature-based surface shape analysis. The curvatures of polynomial surfaces roughly classify facial features into flat, planar, convex, concave and saddle patches. Since grey values seen from the outside represent reflected light, we will find concave functions for convex face parts. This classification facilitates many tasks in automated face analysis, demonstrated in this work by face verification on the polynomial representation. The task of face verification is to verify a face detection by analysing an image of the face. As an extension, we also investigate curvature-based surface shape analysis for images and videos of human bodies, in order to reconstruct the human body skeleton, to detect limbs and to estimate the human pose.

To summarize, the following enumeration gives a clear overview of the main contributions in this work:

- A novel face model where the face is seen as a flexible ellipsoid mask with cutouts for the eyes, the mouth, the nose and the nostrils. The contour pixels and the image intensities of the different facial parts are represented by polynomial surfaces and curves that are convex or concave. The flexibility of the model is obtained by allowing polynomials with a variable degree and a variable approximation error.

- Novel contour and image segmentation algorithms based on adaptive region growing and low-degree polynomial fitting to extract geometric low-level features from contour pixels and image intensities, respectively. These algorithms use a new adaptive thresholding technique with the L_∞ fitting cost as a segmentation criterion. The polynomial degree and the fitting error are automatically adapted during the region growing process. The main novelty is that the algorithms detect outliers, distinguishes between strong and smooth discontinuities and find segments that are bent in a certain way, such as convex or concave segments. Adaptive refers to the use of a local neighbourhood to add pixels, while adapting the shape (or degree) of the function is

based on global behaviour. In this sense there is some local flexibility, while the global behaviour is determined by a more straightforward characterization, such as being concave or convex. This work was published in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- An original solution for the correspondence problem of polynomial curves approximating contours in different images. The main contribution is the introduction of intensity variations in the matching function. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

- A new way of curvature-based surface shape analysis of faces and human bodies in images. The main idea is to use the curvatures of polynomial surfaces to classify facial and human body features into flat, planar, convex, concave and saddle patches. This classification facilitates the analysis of facial and human behaviour. This work was published in [Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- A novel segmented face approximation algorithm to send and store images of faces at a low bit rate, such that the faces are still recognizable and that the compression does not prevent remote face analysis. Segmented face approximation with low-degree polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image. This work was published in [Deboeverie et al., 2013c].

- A practical framework for face analysis applications, e.g. recognition and tracking of faces and facial features. We evaluate the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compare the proposed methods with several techniques of the state of the art. In extension, we apply our algorithms on several other objects, such as vehicles. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2011, Deboeverie et al., 2012].

In total, the research during this PhD resulted in three papers as first author and one paper as second author in international peer-reviewed journals [Deboeverie et al., 2013c, Deboeverie et al., 2013b, Deboeverie et al., 2013a, Bo Bo et al., 2014], of which two are published and two are submitted. Furthermore, ten conference papers as first author were published in the proceedings of international or national conferences [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie, 2008, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2010, Deboeverie et al., 2011, Deboeverie, 2011, Deboeverie et al., 2012, Deboeverie et al., 2014] and three publications as co-author [Geelen et al., 2009, Maes et al., 2009, Eldib et al., 2014].

This work has led to important and critical contributions to the finished projects ISYSS (Intelligent SYstems for Security and Safety) and iCocoon (Immersive

COmmunication by means of COmputer visiON). The experience gained in the projects ISYSS and iCocoon is now being used to contribute in the recently started projects LittleSister (low-cost monitoring for care and retail) and SONOPA (SOcial Networks for Older adults to Promote an Active life).

Samenvatting

In de huidige maatschappij is het gebruik van beeld- en videogegevens wijd verspreid door de toenemende beschikbaarheid van hoogwaardige beeldvormingstoestellen, zoals digitale fotocamera's. Mensen gebruiken beeld- en videodata om hun ervaringen in het dagelijkse leven te delen met andere mensen, bijvoorbeeld op sociale media. Ook directe communicatie op afstand is zeer populair geworden. In geautomatiseerde videoconferencing, bijvoorbeeld, maken intelligente camera's gebruik van gezichtsanalysesystemen met geïntegreerde gezichtsdatabanken. In deze applicatie zal een compacte representatie van het gezicht de verwerkings- en opslagkosten verminderen. In het ideale geval voor video analyse zou een dergelijke representatie gebaseerd zijn op een perfecte segmentatie van het gezicht in al zijn belangrijke onderdelen, namelijk het voorhoofd, de neus, de ogen, de lippen, enzovoort. Het is bijvoorbeeld belangrijk dat de lip nauwgezet de beweging volgt van de spraak met vrijwel geen tijdsvertraging, wat veel gemakkelijker te bereiken is als de beeldrepresentatie de lippen als afzonderlijke delen bevat. Bovendien zal het coderen op basis van een slimme gezichtssegmentatie bijkomende taken eenvoudiger maken, zoals het herkennen van gezichten, gezichtsuitdrukkingen en kijkrichtingen.

Dit doctoraatsonderzoek brengt een oplossing voor de segmentatie, approximatie en analyse van gezichten in visuele communicatietoepassingen met behulp van computervisie.

- **Segmentatie van gezichten** is de verdeling van gezichtsafbeeldingen in betekenisvolle fysieke onderdelen, zoals het voorhoofd, de wangen, de lippen, de wenkbrauwen, enzovoort. In dit werk stellen we voor om contourpixels en beeldpixels (grijswaarden, beeldintensiteiten) in gezichtsafbeeldingen te groeperen met gebiedsuitbreiding en polynomiale fitting. Gebiedsuitbreiding is het proces van het onderzoeken van naburige pixels van initiële zaaipixels en het bepalen of de naburige pixels moeten worden toegevoegd aan het gebied. Polynomiale fitting past geometrische primitieven aan pixels. Een geometrische primitieve is een polynomiale functie die de geometrie van een rand of de variatie van de grijswaarden in een gebied beschrijft. Het probleem dat we dus bestuderen is het vinden van een gebied van maximale grootte waar contourpixels enerzijds en grijswaarden anderzijds goed kunnen worden benaderd door een polynomiale functie. Segmentatie van gezichten in dit werk beschouwt het groeperen van contourpixels in contoursegmenten, evenals het groeperen van gezichtsintensiteiten in oppervlaktesegmenten.

Om segmenten te vinden stellen we een adaptief algoritme voor van gebiedsuitbreiding gebaseerd op constructieve polynomiale fitting. Dit algoritme voor extractie van primitieven vindt deelverzamelingen van pixels die op of dichtbij een geometrische primitieve liggen. Hoe goed een subset overeen komt met een primitieve wordt gemeten door een L_∞ fitting kost (benaderingsfout). Een originele bijdrage is de invoering van adaptieve thresholding voor gebiedsuitbreiding, die een variabele polynomiale graad en een variabele fitting kost toelaat, afhankelijk van de lokale eigenschappen van de pixels. De L_∞ fitting kost is de maximale afwijking tussen de pixels en de polynomial functie. De vernieuwing is dat de gebiedsuitbreiding outliers detecteert, onderscheid maakt tussen sterke en zachte discontinuïteiten en de kromming beschouwt, zoals convexiteit of concaviteit. Een outlier is een pixel die afwijkt van zijn buren vanwege ruis of kleine object spikkels. De gebiedsuitbreiding onderzoekt de lokale variatie van pixels in een segment om outliers te identificeren en de globale variatie van pixels in een segment wordt onderzocht om de graad van de polynomiale functie aan te passen. De termen lokaal en globaal slaan hier respectievelijk op een deel van het segment en het volledige segment. De combinatie van beide is mogelijk omdat we gebruik maken van constructieve fitting: de globale fitting kost wordt berekend op basis van de lokale fitting kosten. In dit werk tonen we aan hoe lokale en globale fitting kosten kunnen interageren met elkaar in een adaptieve methode.

Terwijl voor de segmentatie van gezichten de focus van het onderzoek ligt op het groeperen van de pixels, focust de appoximatie van gezichten op het produceren van nauwkeurige benaderingen van de pixels.

- **Approximatie van gezichten** is de schatting en de reconstructie van contourpixels en grijswaarden in gezichtsafbeeldingen naar een gewenste mate van nauwkeurigheid. In dit werk stellen we voor om gezichten te benaderen met mappen gedefinieerd door polynomiale functies (geometrische laagniveau features, geometrische primitieven) van lage graad (bv. 0, 1 of 2). Een moeilijkheid die we bestuderen is het vinden van de parameters van de beste fit.

  Het contourmodel verzamelt contoursegmenten als polynomiale curven, die ofwel recht, convex of concaaf zijn, in een Curve Edge Map (CEM). Het oppervlaktemodel verzamelt oppervlaktesegmenten als polynomiale oppervlakken, die ofwel plat, vlak, convex, concaaf of een zadelvormig zijn, in een Surface Intensity Map (SIM). Beide modellen zijn eenvoudig, natuurlijk, handig en elegante representaties voor objecten in beelden, in het bijzonder voor gezichtsafbeeldingen. De polynomiale functies van lage graad in deze modellen zorgen voor een goede benadering van zinvolle gezichtsfeatures, terwijl alle noodzakelijke details van het gezicht in het gereconstrueerde beeld behouden blijven.

  De contour- en oppervlaktemodellen laten toe om een gezicht te parametri-

seren aan de hand van de coëfficiënten van de polynomiale curven en oppervlakken en de coördinaten van de eindpunten van de curven in een paar honderd bytes. Het versturen van deze gezichtsparameters via een netwerk is zeer efficiënt en de methode behoudt alle noodzakelijke details van het gezicht in het gereconstrueerde beeld. Bovendien zijn deze gezichtsdescriptoren geschikt voor geautomatiseerde gezichtsanalyse.

- **Gezichtsanalyse** omvat het herkennen en het tracken van gezichten en gezichtsfeatures. Gezichtsherkenning identificeert een persoon in een digitale foto of een videobeeld. Gezichtstracking volgt de bewegingen van het hoofd van een persoon in een video. Herkenning en tracking van gezichten en gezichtsfeatures leidt tot de detectie van specifiek gezichtsgedrag, zoals spreken, de kijkrichting, hoofdbewegingen (bijvoorbeeld knikken) of gezichtsuitdrukkingen (bijvoorbeeld blij/verdrietig/boos/verbaasd).

  Om gezichten in twee verschillende of opeenvolgende beelden te vergelijken, presenteren we een techniek om correspondenties van geometrische functies te vinden. Het doel van een algoritme voor correspondenties of matching is om aan te geven voor een punt (feature) in een eerste afbeelding wat het overeenkomstige punt (feature) is in een tweede afbeelding, waarbij beide beeldpunten hetzelfde 3-D wereldpunt moeten aanduiden.

  In het contourmodel vinden we correspondenties door een techniek die polynomiale curven matcht op basis van vorm, relatieve positie en intensiteit. We stellen een ongelijkheidsfunctie voor om lokaal curven te matchen, evenals een gelijkheidsfunctie om globaal curven te matchen. Het verschil zit in de toepassing: lokaal matchen wordt vooral gebruikt bij tracking toepassingen, terwijl globaal matchen zich richt op herkenningstoepassingen.

  In het oppervlaktemodel analyseren we de vorm van oppervlakken gebaseerd op kromming. De krommingen van de polynomiale oppervlakken classificeren gezichtsfeatures ruwweg in platte, vlakke, convexe, concave en zadelvormige delen. Omdat grijswaarden gezien vanaf de buitenkant gereflecteerd licht voorstellen, zullen we concave functies vinden voor convexe gezichtsdelen. Deze classificatie maakt veel taken in geautomatiseerde gezichtsanalyse eenvoudiger, aangetoond in dit werk door gezichtsverificatie op de polynomiale representatie. De taak van het gezichtsverificatie is het controleren van een gezichtsdectectie door het analyseren van een gezichtsafbeelding. Als uitbreiding onderzoeken we de kromming van oppervlakken van het menselijk lichaam in afbeeldingen en video's, om het skelet te reconstrueren en zo ledematen te detecteren en de houding van het lichaam in te schatten.

Om samen te vatten geeft de volgende opsomming een duidelijk overzicht van de belangrijkste bijdragen in dit werk:

- Een nieuw gezichtsmodel waarbij het gezicht gezien wordt als een flexibel ellipsoïdaal masker met uitsparingen voor de ogen, de mond, de neus and de neusgaten. De contourpixels en de beeldintensiteiten van de verschillende gezichtsonderdelen worden voorgesteld met polynomiale oppervlakken en curven die convex of concaaf zijn. De flexibiliteit van het model wordt bekomen door het toelaten van polynomen met een variabele graad en een variabele benaderingsfout.

- Nieuwe algoritmen voor segmentatie van contouren en beelden gebaseerd op adaptieve gebiedsuitbreiding en polynomiale fitting van lage graad om geometrische laagniveau features te extraheren uit respectievelijk contourpixels en beeldintensiteiten. Deze algoritmen maken gebruik van een nieuwe adaptieve thresholding techniek met de L_∞ fitting kost als een criterium voor segmentatie. De polynomiale graad en de fitting fout worden automatisch in het gebiedsuitbreidingsproces aangepast. De belangrijkste vernieuwing is dat de algoritmen outliers detecteren, onderscheid maken tussen sterke en zachte discontinuïteiten, en segmenten vinden die gebogen zijn op een bepaalde manier, zoals convexe of concave segmenten. Adaptief verwijst naar het gebruik van een lokale omgeving om pixels toe te voegen, terwijl het aanpassen van de vorm (of de graad) van een functie gebaseerd is op globaal gedrag. In deze betekenis is er enige lokale flexibiliteit, terwijl het globaal gedrag bepaald is door bijvoorbeeld een concave of convexe karakterisering. Dit werk werd gepubliceerd in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- Een originele oplossing voor het correspondentieprobleem bij polynomiale curven die de contouren in verschillende afbeeldingen benaderen. De belangrijkste bijdrage is de invoering van intensiteitvariaties in de matchingfunctie. Dit werk werd gepubliceerd in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

- Een nieuwe manier van analyse van de vorm van oppervlakken, gebaseerd op kromming, in afbeeldingen van gezichten en menselijke lichamen. Het voornaamste idee is om de krommingen van de polynomial oppervlakken te gebruiken om gezichtsfeatures en lichaamsfeatures in te delen in platte, vlakke, convexe, concave en zadelvormige delen. Deze indeling maakt de analyse van het gezichtsgedrag en het menselijk lichaamsgedrag eenvoudiger. Dit werk werd gepubliceerd in [Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- Een nieuw algoritme voor gesegmenteerde gezichtsbenadering om afbeeldingen van gezichten te versturen en op te slaan bij een lage bitsnelheid, zodat de gezichten nog steeds herkenbaar zijn en de zodat de compressie

gezichtsanalyse op afstand niet verhindert. Gesegmenteerde gezichtsbenadering met polynomiale oppervlakken en curven van lage graad is heel natuurlijk en biedt een compacte en reversibele manier om de essentiële kenmerken van de oorspronkelijke gezichtsafbeelding te behouden. Dit werk werd gepubliceerd in [Deboeverie et al., 2013c].

- Een praktisch framework voor toepassingen met gezichtsanalyse, bijvoorbeeld het herkennen en het volgen van gezichten en gezichtsfeatures. We evalueren de prestaties van de toepassingen met gezichtsanalyse op een groot aantal representatieve databanken en videosequenties. Verder vergelijken we de voorgestelde methoden met de technieken uit de state of the art. In uitbreiding passen we onze algoritmen toe op verschillende andere objecten, zoals voertuigen. Dit werk werd gepubliceerd in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2011, Deboeverie et al., 2012].

In totaal heeft het onderzoek tijdens deze doctoraatsstudie geresulteerd in drie papers als eerste auteur en één paper als tweede auteur in internationale peer-reviewed tijdschriften [Deboeverie et al., 2013c, Deboeverie et al., 2013b, Deboeverie et al., 2013a, Bo Bo et al., 2014], waarvan er twee gepubliceerd werden en waarvan er twee ingediend zijn. Verder zijn er tien conferentie papers als eerste auteur gepubliceerd in de proceedings van internationale of nationale conferenties [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie, 2008, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2010, Deboeverie et al., 2011, Deboeverie, 2011, Deboeverie et al., 2012, Deboeverie et al., 2014] en drie publicaties als coauteur [Geelen et al., 2009, Maes et al., 2009, Eldib et al., 2014].

Dit werk heeft geleid tot belangrijke en kritische bijdragen in de reeds beeindigde projecten ISYSS (Intelligent SYstems for Security and Safety) en iCocoon (Immersive COmmunication by means of COmputer visiON). De ervaring die is opgedaan in de projecten Isyss en iCocoon wordt nu gebruikt voor bijdragen in de onlangs gestarte projecten LittleSister (low-cost monitoring for care and retail) en SONOPA (SOcial Networks for Older adults to Promote an Active life).

Contents

Figures

Tables

Acronyms

AAL	Ambient Assisted Living
AAM	Active Appearance Model
ACPF	Adaptive constructive polynomial fitting
ASIFT	Affine-Scale-Invariant Feature Transform
ASM	Active Shape Model
BAM	Boosted Appearance Model
BSDS	Berkeley Segmentation DataSet
BSP	Binary Space Partitioning
CEM	Curve Edge Map
CHT	Circle Hough Transform
CLM	Constrained Local Model
DSS	Digital Straight Segments
EBGM	Elastic Bunch Graph Matching
FAST	Features from Accelerated Segment Test
FN	False Negative
FP	False Positive
FPGA	Field-Programmable Gate Array
GTFD	Georgia Tech Face Database
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
iCocoon	Immersive COmmunication by means of COmputer visiON
IP	Implicit Polynomial
ISYSS	Intelligent SYstems for Security and Safety
JPEG	Joint Photographic Experts Group
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LEM	Line Edge Map
LSD	Line Segment Distance
MPEG	Moving Picture Experts Group
MSER	Maximally Stable Extremal Regions
NACPF	Non-adaptive constructive polynomial fitting
NURBS	Non-Uniform Rational B-Splines
PCA	Principal Component Analysis
PSC	Polynomial surfaces and curves
PSNR	Peak Signal-to-Noise Ratio

PRI	Probabilistic Rand Index
PSC	Polynomial Surfaces and Curves
PWS	Power Watersheds
QB	Quad-Binary
RANSAC	RANdom SAmple Consensus
RBC	Red Blood Cell
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SIC	Segmented Image Coding
SIFT	Scale-Invariant Feature Transform
SIM	Surface Intensity Map
SIMD	Single Instruction Multiple Data
SLIC	Sub-image based Lossy Image Compression
SONOPA	SOcial Networks for Older adults to Promote an Active life
SUSAN	Smallest Univalue Segment Assimilating Nucleus
SVD	Singular Value Decomposition
TP	True Positive
TN	True Negative
VQ	Vector Quantization
VAD	Voice Activity Detection
YFD	Yale Face Database

1 Introduction

1.1 Content

In this introductory chapter, we situate the research presented in this PhD thesis. Computer vision includes the development of intelligent methods for acquiring, processing, analysing, and understanding images and videos, in order to produce numerical or symbolic information, e.g., in the forms of decisions. Computer vision aims to imitate and extend the complex human brain, which understands and interprets images captured by the human eye. For instance, several models have been proposed that attempt to explain how the brain identifies people by looking at their faces [Martínez, 2003]. Computer vision applications make our lives more comfortable and safer, for instance with automated face recognition in surveillance systems. In this work, we use computer vision to solve important problems about recognizing human identity and human behaviour in visual communication applications.

A toy example for the work in this dissertation is automated video conferencing. Video conferencing implies communication between two or more locations by simultaneous two-way video and audio transmissions. In automated video conferencing, persons in a meeting are observed by several cameras. Figure 1.1 shows an example of a meeting room setup for automated video conferencing. In a meeting, observed with several cameras, we use computer vision to find the identity of the participants and to analyse their behaviour. Behaviour includes activities, such as entering/leaving, speaking, interaction with other persons, as well as emotions,

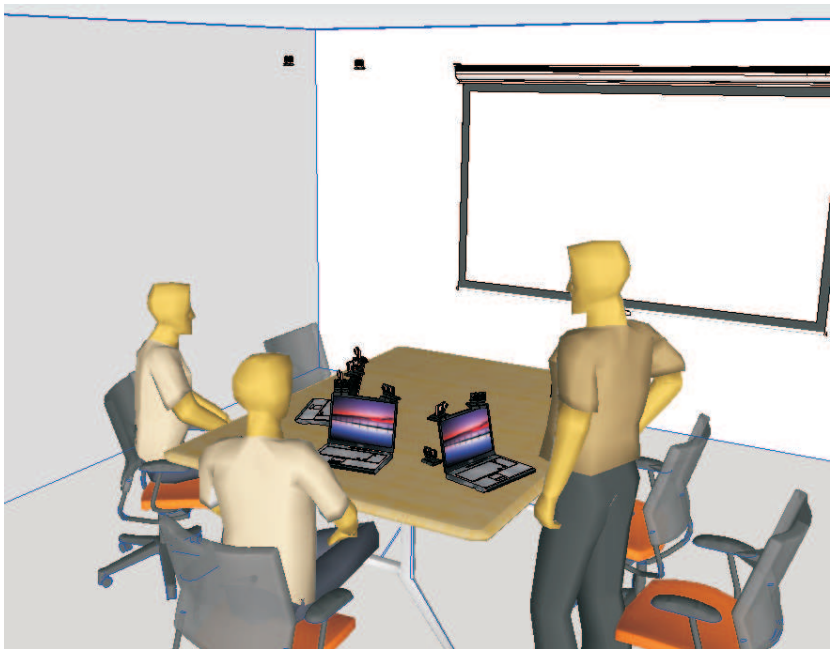


Figure 1.1: Example of a meeting room setup for automated video conferencing.

such as being happy/sad. Related problems are the detection of the participants' viewing direction and selecting the camera with the best view on a person. Solving these problems with computer vision demands a compact face image representation to transmit, store, visualize and analyse faces efficiently. Thus, our final purpose in this work is to develop a system for face segmentation, approximation and analysis. Therefore, we propose contour and surface face models to segment, approximate and analyse images and videos of faces.

- **Face segmentation** is the division of face images into physically meaningful parts, such as the forehead, the cheeks, the lips, the eyebrows, etc. In this work, we propose to group contour pixels and image pixels (grey values, image intensities) in images of faces with region growing and polynomial fitting. Region growing is the process of examining neighbouring pixels of initial seed pixels and determining whether the neighboring pixels should be added to the region. Polynomial fitting fits geometric primitives to pixels. A geometric primitive is a polynomial function describing the geometry of an edge or the variation of grey values in a region. Thus, the problem we study is that of finding a region of maximal size in which grey values can be well approximated by a polynomial function and where contour pixels

(a) (b) (c)

Figure 1.2: (a): A greyscale face image in the Georgia Tech Face Database [GTFD,]. (b): The face contours obtained with Canny edge detection [Canny, 1986]. (c): The face contours segmented into polynomial curves of second degree.

can be well approximated by polynomials as well. Face segmentation in this work considers the grouping of facial contour pixels into contour segments, as well as the grouping of facial intensities into surface segments.

To find segments, we propose an adaptive region growing algorithm based on constructive polynomial fitting [Veelaert, 1997, Veelaert and Teelen, 2006, Veelaert, 2012]. This primitive extraction algorithm finds subsets of pixels that lie on a geometric primitive or close to it. How well a subset corresponds to a primitive is quantified by an L_∞ fitting cost (approximation error). An original contribution is the introduction of adaptive thresholding for region growing, which allows a variable polynomial degree and a variable fitting cost, depending on the local properties of the pixels. The L_∞ fitting cost is the maximum deviation between the pixels and the polynomial function. The novelty is that region growing detects outliers, distinguishes between strong and smooth discontinuities and considers the curvature, such as convexity or concavity. An outlier is a pixel differing from its neighbours due to noise or small object speckles. The region growing investigates the local variation of pixels in a segment to identify outliers, while the global variation of pixels in a segment is investigated to adapt the degree of the polynomial function. The terms local and global refer to a part of the segment and the entire segment, respectively. The combination of both is possible because we employ constructive fitting: the global fitting cost is calculated from local fitting costs. In this work, we demonstrate how local and global fitting costs can interact in an adaptive method. Figure 1.2 illustrates an example of the segmentation of facial contours into polynomial curves of second

(a) (b) (c)

Figure 1.3: (a): A greyscale face image of the Stirling face database [Stirling,]. (b): The face image intensities segmented into surface segments. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. (c): The surface segments approximated by low-degree polynomial surfaces.

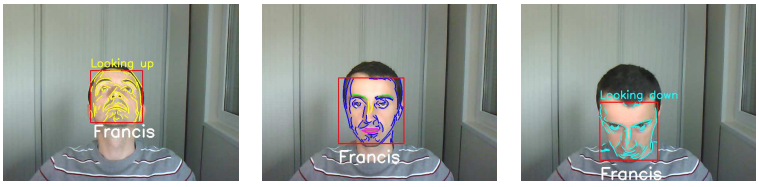
degree.

Whereas in face segmentation the focus of the research is on grouping the pixels, face approximation focuses on producing accurate approximations of the pixels.

- **Face approximation** is the estimation and the reconstruction of contour pixels and grey values of face images to a desired degree of accuracy. In this work, we propose to approximate faces with maps defined by polynomial functions (geometric low-level features, geometric primitives) of low-degree (e.g., 0, 1 or 2). One difficulty we study is that of finding the parameters of the best fit.

  The contour model represents contour segments with polynomial curves, which are either straight, convex or concave, in a Curve Edge Map (CEM). The surface model represents surface segments as polynomial surfaces, which are either flat, planar, convex, concave or saddle surfaces, in a Surface Intensity Map (SIM). Both models are simple, natural, useful and elegant representations for objects in images, in particular for face images. The low-degree polynomial functions in these models provide good approximation of meaningful facial features, while preserving all the necessary details of the face in the reconstructed image.

  The contour and surface models allow parametrizing a face by the coefficients of the polynomial curves and surfaces and the coordinates of the endpoints of the curves in a few hundred bytes. Transmitting these face pa-



(a) (b) (c)

Figure 1.4: The images in Figures (a), (b) and (c) show a face from a webcam video sequence, which is looking up, frontal and right, respectively. Here, face analysis includes face recognition, head movements detection (nodding) and facial features classification.

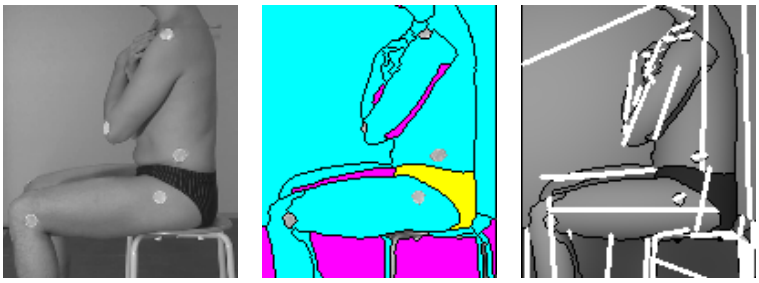
rameters over the network is very efficient and the method preserves all the necessary details of the face in the reconstructed image. Furthermore, these face descriptors are suitable for automated face analysis. Figure 1.3 illustrates an example of the approximation of facial intensities by low-degree polynomial surfaces.

- **Face analysis** includes recognition and tracking of faces and facial features. Face recognition identifies a person in a digital image or a video frame. Face tracking follows the movements of a persons head in a video. Recognition and tracking of faces and facial features leads to the detection of specific human face behaviour, such as speaking, gaze direction, head movements (e.g. nodding) or facial expressions (e.g. happy/sad/angry/surprised). Figure 1.4 shows an example of face recognition, head movement detection and facial feature classification of a face from a webcam video sequence.

  In order to compare faces in two different or consecutive images, we present a technique to find geometric feature correspondence pairs. The goal of a correspondence finding or matching algorithm is to indicate for a point (feature) in one image which is the corresponding point (feature) in a second image, where both image points must show the same 3-D world point.

  In the contour model, we find correspondences by a technique which matches polynomial curves, based on shape, relative position and intensity. We propose a dissimilarity function for local curve matching as well as a similarity function for global curve matching. The difference lies in the application: local matching is primarily used in tracking applications, while global matching focuses on recognition applications.

  In the surface model, we perform curvature-based surface shape analysis. The curvatures of polynomial surfaces roughly classify facial features into flat, planar, convex, concave and saddle patches. Since grey values seen



(a) (b) (c)

Figure 1.5: (a): A greyscale human body image. (b): The segmented human body. The convex, concave or saddle like behaviour of the polynomial surfaces, indicated by the colours magenta, cyan and yellow, respectively. (c): Human body skeleton reconstruction.

from the outside represent reflected light, we will find concave functions for convex face parts [Wagemans et al., 2010]. This classification facilitates many tasks in automated face analysis, demonstrated in this work by face verification on the polynomial representation. The task of face verification is to verify a face detection by analysing an image of the face. As an extension, we also investigate curvature-based surface shape analysis for images and videos of human bodies, in order to reconstruct the human body skeleton, to detect limbs and to estimate the human pose. Figure 1.5 shows an example of curvature-based surface shape analysis of a human body.

1.2 Contributions

Todays most performant techniques in computer vision attach importance to the way of summarizing the local image structure, whereby scale is a major focus. For instance, the face detector of Viola and Jones [Viola and Jones, 2001] finds faces by repeating the classification at different scales. Another example is the Scale-Invariant Feature Transform (SIFT) of Lowe [Lowe, 2004]. The SIFT method returns scale invariant feature point locations based on measures derived directly from local sampling of the intensity values. Another example is the use of Histograms of Oriented Gradients (HOG) to detect pedestrians by Dalal and Triggs [Dalal and Triggs, 2005]. The method counts occurrences of gradient orientations in local image regions. Human detection performance is further improved by combining HOG with Local Binary Patterns (LBP) [W. Xiaoyu, 2009]. These features have also been employed in face recognition algorithms [Zhang et al., 2010, Déniz et al., 2011, Geng and Jiang, 2013]. A final example is the

keypoint descriptor inspired by the human visual system and more precisely the retina, coined Fast Retina Keypoint (FREAK) [Alahi et al., 2012]. This descriptor computes a cascade of binary strings by comparing image intensities over a retinal sampling pattern. The basic idea of SIFT, HOG, LBP and FREAK is to analyse the local structure in an image by comparing each pixel with its neighbourhood.

With this trend in mind, the purpose of this work is to show that older techniques, such as polynomial segmentation, can still compete with state-of-the-art techniques, if we combine them with clever techniques that make use of scale and adaptive techniques.

The following overview summarizes the main contributions in this work:

- A novel face model where the face is seen as a flexible ellipsoid mask with cutouts for the eyes, the mouth, the nose and the nostrils. The contour pixels and the image intensities of the different facial parts are represented by polynomial surfaces and curves that are convex or concave. The flexibility of the model is obtained by allowing polynomials with a variable degree and a variable approximation error.

- Novel contour and image segmentation algorithms based on adaptive region growing and low-degree polynomial fitting to extract geometric low-level features from contour pixels and image intensities, respectively. These algorithms use a new adaptive thresholding technique with the L_∞ fitting cost as a segmentation criterion. The polynomial degree and the fitting error are automatically adapted during the region growing process. The main novelty is that the algorithms detect outliers, distinguish between strong and smooth discontinuities and find segments that are bent in a certain way, such as convex or concave segments. Adaptive refers to the use of a local neighbourhood to add pixels, while adapting the shape (or degree) of the function is based on global behaviour. In this sense there is some local flexibility, while the global behaviour is determined by a more straightforward characterization, such as being concave or convex. This work has been published in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- An original solution for the correspondence problem of polynomial curves approximating contours in different images. The main contribution is the introduction of intensity variations in the matching function. This work has been published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

- A new way of curvature-based surface shape analysis of faces and human bodies in images. The main idea is to use the curvatures of polynomial surfaces to classify facial and human body features into flat, planar, convex,

concave and saddle patches. This classification facilitates the analysis of facial and human behaviour. This work has been published in [Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- A novel segmented face approximation algorithm to send and store images of faces at a low bit rate, such that the faces are still recognizable and that the compression does not prevent remote face analysis. Segmented face approximation with low-degree polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image. This work has been published in [Deboeverie et al., 2013c].

- A practical framework for face analysis applications, e.g. recognition and tracking of faces and facial features. We evaluate the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compare the proposed methods with several techniques of the state of the art. In extension, we apply our algorithms on several other objects, such as vehicles. This work has been published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2011, Deboeverie et al., 2012].

1.3 Outline

This section gives an overview of the work and the different chapters in this thesis, as visualised in Figure 1.6.

- **Chapter 2: Contour and surface models**
  In Chapter 2, we propose algorithms that use adaptive region growing and polynomial fitting to group pixels into segments and to approximate them by polynomial functions. We investigate adaptive region growing for contour segmentation as well as for image segmentation. In the 1-D case, we firstly group contour pixels into contour segments. Then, we represent the contour segments by polynomial curves, which are either straight, convex or concave. In the 2-D case, we firstly segment image intensities into surface segments (intensity patches). Then, we represent the surface segments as polynomial surfaces, that are either flat, planar, convex, concave or behave like saddle surfaces. The polynomial curves and surfaces are grouped into two novel compact features: the Curve Edge Map (CEM) and the Surface Intensity Map (SIM), respectively. Adaptive region growing in both cases is based on the same principles of local and global sampling of the region. However, mainly due to the different spatial ordering of contour

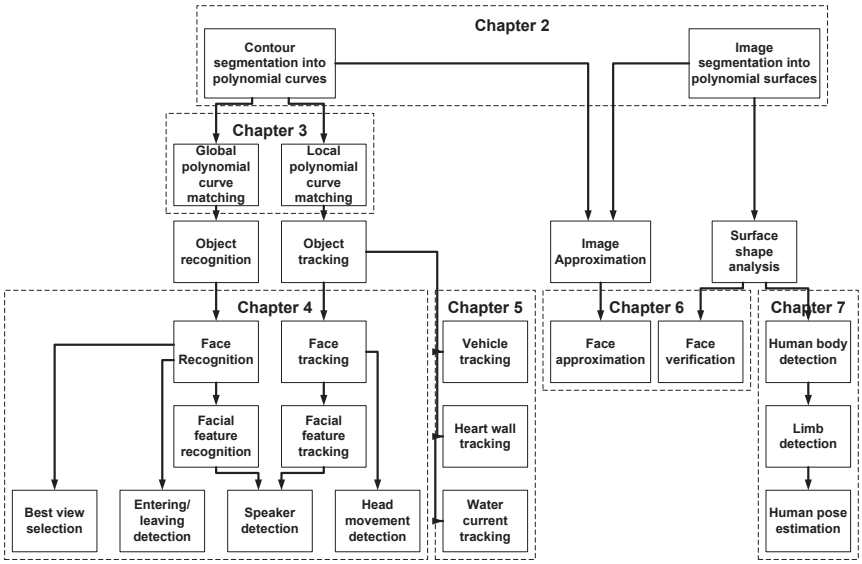


Figure 1.6: Block diagram with an overview of the different chapters in this thesis

pixels and image pixels, both need their own modifications of adaptive region growing. Where region growing groups pixels along one dimension for connected contour pixels, region growing groups pixels along two dimensions for connected image pixels. In both cases, this requires different strategies when selecting elemental subsets. Moreover, different polynomial fitting functions for contour regions and image regions imply different computations for constructive fitting. This work was published in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- **Chapter 3: Polynomial curve matching**
  In Chapter 3, we find correspondences in Curve Edge Maps (CEMs), as proposed in Chapter 2, by a technique which matches polynomial curves, based on shape, relative position and intensity. We propose a dissimilarity function for local curve matching as well as a similarity function for global curve matching. The difference lies in the application: local matching is especially used in tracking applications, while global matching focuses on recognition applications. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

- **Chapter 4: Face analysis applications**
  In Chapter 4, we evaluate the matching techniques for polynomial curves, as proposed in Chapter 3, by face recognition and tracking. Here, we consider a top-down approach. Firstly, faces are recognized and tracked by matching all geometric features in face images. Then, individual facial features are classified and further analysed. The applications considered are people identification, best view selection and behaviour analysis applications, such as entering/leaving detection, head movement detection and speaker detection. We evaluate the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compare the proposed methods with several techniques of the state of the art. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2011, Deboeverie et al., 2012].

- **Chapter 5: Tracking of other objects**
  In Chapter 5, we evaluate the matching techniques for polynomial curves, as proposed in Chapter 3, by tracking of other objects than faces, such as vehicles, heart walls and water currents. This work was published in [Deboeverie et al., 2009a, Deboeverie et al., 2009b].

- **Chapter 6: Applications of segmented face approximation**
  In Chapter 6, the main purpose is to obtain a complete face representation. We group face image intensities into meaningful surface segments and approximate them with maps defined by low-degree polynomial surfaces (SIMs), as proposed in Chapter 2. Then, we segment the contours separating the surface segments into contour segments and represent them with maps defined by low-degree polynomial curves (CEMs), as proposed in Chapter 2. We also examine curvature-based surface shape analysis for face images, demonstrated by face verification on the polynomial representation. This work was published in [Deboeverie et al., 2013c].

- **Chapter 7: Human body analysis**
  In Chapter 7, we aim to reconstruct the human body skeleton, to detect limbs and to estimate the human pose. Therefore, we segment greyscale images of human bodies into smooth surface segments, as proposed in Chapter 2. Then, we approximate these human body parts by nearly cylindrical surfaces, of which the axes of minimum curvature accurately reconstruct the human body skeleton. This work was submitted in [Deboeverie et al., 2013a].

1.4 Publications

In total, the research during this PhD resulted in three papers as first author and one paper as second author in international peer-reviewed journals [Deboeverie et al., 2013c, Deboeverie et al., 2013b, Deboeverie et al., 2013a, Bo Bo et al., 2014], of which two are published and two are submitted. Furthermore, ten conference papers as first author have been published in the proceedings of international or national conferences [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie, 2008, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2010, Deboeverie et al., 2011, Deboeverie, 2011, Deboeverie et al., 2012, Deboeverie et al., 2014] and three publications as co-author [Geelen et al., 2009, Maes et al., 2009, Eldib et al., 2014].

1.4.1 Publications in international journals

- [Deboeverie et al., 2013c] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Segmented Face Approximation with Adaptive Region Growing based on Low-degree Polynomial Fitting, Signal, Image and Video Processing, 2013.

- [Deboeverie et al., 2013b] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Image Segmentation with Adaptive Region Growing based on a Polynomial Surface Model, Electronic Imaging, 2013.

- [Deboeverie et al., 2013a] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Human body parts segmentation in physiotherapy with adaptive curvature-based region growing, International Journal of Computer Vision, 2013, submitted.

- [Bo Bo et al., 2014] Nyan Bo Bo, Francis Deboeverie, Mohamed El Dib, Junzhi Guan, Xingzhe Xie, Jorge Niño Casteñada, Dirk Van Haerenborgh, Maarten Slembrouck, Samuel Van de Velde, Heidi Steendam, Peter Veelaert, Richard Kleihorst, Hamid Aghajan and Wilfried Philips, Human Mobility Monitoring in Very Low-Resolution Visual Sensor Network, MDPI special issue on Ambient Assisted Living (AAL): Sensors, Architectures and Applications, 2014, submitted.

1.4.2 Publications in international conferences

- [Deboeverie et al., 2008b] Francis Deboeverie, Peter Veelaert, Kristof Teelen and Wilfried Philips, Face Recognition Using Parabola Edge Map, Advanced Concepts for Intelligent Vision Systems, 2008.

- [Deboeverie et al., 2008a] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Parabola-based Face Recognition and Tracking, ProRISC 2008.
- [Maes et al., 2009] Frédéric Maes, Francis Deboeverie, Bill Chaudhry, Peter Van Ransbeeck, Pascal Verdonck, Tools to understand the pumping mechanism of embryonic hearts, International Conference on Computational Biology, 2009.
- [Deboeverie et al., 2009a] Francis Deboeverie, Frédéric Maes, Peter Veelaert and Wilfried Philips, Linked Geometric Features for Modeling the Fluid Flow in Developing Embryonic Vertebrate, International Conference on Image Processing, 2009.
- [Deboeverie et al., 2009b] Francis Deboeverie, Kristof Teelen, Peter Veelaert and Wilfried Philips, Rigid Object Tracking Using Geometric Features, Advanced Concepts for Intelligent Vision Systems, 2009.
- [Geelen et al., 2009] Bert Geelen, Francis Deboeverie and Peter Veelaert, Mapping of Parabola Edge Map Canny Preprocessing to the Xetal SmartCam Architecture, International Conference on Distributed Smart Cameras, 2009.
- [Deboeverie et al., 2010] Francis Deboeverie, Kristof Teelen, Peter Veelaert and Wilfried Philips, Adaptive Constructive Polynomial Fitting, Advanced Concepts for Intelligent Vision Systems, 2010.
- [Deboeverie et al., 2011] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Face Analysis using Curve Edge Maps, International Conference on Image Analysis and Processing, 2011.
- [Deboeverie et al., 2012] Francis Deboeverie, Peter Veelaert and Wilfried Philips, Best View Selection with Geometric Feature based Face Recognition, International Conference on Image Processing, 2012.
- [Deboeverie et al., 2014] Francis Deboeverie, Gianni Allebosch, Dirk Van Haerenborgh, Peter Veelaert and Wilfried Philips, Edge-based Foreground Detection with Higher Order Derivative Local Binary Patterns for Low-resolution Video Processing, International Conference on Computer Vision Theory and Applications, 2014.
- [Eldib et al., 2014] Mohamed Eldib, Nyan Bo Bo, Francis Deboeverie, Jorge Niño Castañeda, Junzhi Guan, Samuel Van de Velde, Heidi Steendam, Hamid Aghajan and Wilfried Philips, A Low Resolution Multi-Camera System For Person Tracking, International Conference on Image Processing, 2014.

1.4.3 Publications in national conferences

- [Deboeverie, 2008] Francis Deboeverie, Shape matching with geometric primitives, 9th FirW Doctoral Symposium, 2008.

- [Deboeverie, 2011] Francis Deboeverie, A Uniform Approach for Face Segmentation and Coding with Adaptive Region Growing, 12th FirW Doctoral Symposium, 2011.

1.5 Research activities

This work has led to important and critical contributions to the finished projects ISYSS and iCocoon. The experience gained in the projects ISYSS and iCocoon is now being used to contribute in the recently started projects LittleSister and SONOPA.
Contributions to projects:

- **ISYSS:** Intelligent SYstems for Security and Safety [1]
  iMinds GBO project: 1 January 2008 - 31 December 2009
  Consortium: UGent/TELIN/IPI, Barco, NXP, Tele Atlas, Vito, Z-monitoring, UGent-MMLab, VUB-ETRO, Imec-NES.

  Brief description: During crisis situations, emergency services have to make many crucial decisions based on data gathered from various sources. Wireless mobile camera networks are a good solution for setting up emergency information networks, as they do not rely on fixed infrastructure, can be set up quickly, and provide the necessary communication bandwidth. However, the information overload that is generated from cameras on land and airborne vehicles and surveillance cameras in the field, poses serious challenges for operators in a control room to maintain a clear overview of a situation and to quickly respond to crisis events. In complex cases, often 50+ cameras need to be monitored putting severe strain on the operator. There is a clear need for automated support in the management of the multitude of videostreams in order to make split second decisions. ISYSS addressed this problem by focusing on intelligent information extraction and management of video data from mobile camera-platforms in crisis management. Advanced processing rendered the video information location and context aware.

  The work in this thesis contributed to ISYSS by means of polynomial curve extraction [Deboeverie et al., 2010], polynomial curve matching [Deboev-

[1] More details can be found at `http://www.iminds.be/en/research/overview-projects/p/detail/isyss`

erie et al., 2008b] and tracking and recognition of rigid objects, such as vehicles [Deboeverie et al., 2009b].

- **iCocoon:** Immersive COmmunication by means of COmputer visiON [2]
  iMinds ICON project: 1 January 2010 - 31 December 2011
  Consortium: Alcatel-Lucent Bell, VITO NV, Eyetronics, UGent/TELIN/IPI, UGent/ELIS/MMLab, VUB/ETRO, VUB/SMIT, KULeuven/PSI-VISICS.

  Brief description: The purpose of iCocoon was to drastically change the way people communicate remotely. This was realized by creating third-generation video conferencing applications based on world-class video technologies (such as Computer Vision, Scene Understanding and 3D). iCocoon provides a much better immersive sensation to the communicating partners and better understand and take into account the context in which the communication is taking place.

  The work in this thesis contributed to iCocoon by means of polynomial curve extraction [Deboeverie et al., 2010], polynomial curve matching [Deboeverie et al., 2008b] and applications by analysis of faces, such as people identification, best view selection, entering/leaving detection, head movement detection and speaker detection [Deboeverie et al., 2008b, Deboeverie et al., 2009b, Deboeverie et al., 2011]. A second part of the contribution included the segmentation, approximation and analysis of faces with polynomial surfaces [Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- **LittleSister:** low-cost monitoring for care and retail [3]
  iMinds ICON project: 1 January 2013 - 31 December 2014
  Consortium: Xetal, NIKO, Seris, JFOceans, CM, UGent-IPI, UGent-MMLab, UGent-MICT, VUB-ETRO, UA-PATS.

  Brief description: Many elderly citizens, even though affected by chronic disabilities, wish to retain their autonomy and enjoy their own home for as long as possible. This leads to a need for Electronics and ICT systems capable of detecting alarming situations that require intervention, or collecting data to anticipate complications in domestic health care. The general idea is to create a cheap and low maintenance sensor system for monitoring behaviour of elderly in their homes, with minimal intrusion on privacy and minimal cabling for ease of installation, to develop distributed algorithms for processing and transmitting data in this sensor network, and to correlate behavioural changes with changes in health. An important challenge is to minimize power usage in the system. This is achieved using a cascaded

[2] More details can be found at `http://www.iminds.be/en/research/overview-projects/p/detail/icocoon-2`

[3] More details can be found at `http://www.iminds.be/en/research/overview-projects/p/detail/littlesister`

approach. Very low power sensors are active all the time; some of the low resolution sensors are activated frequently, but then rely on sophisticated distributed video processing algorithms to avoid power hungry video transmission. At crucial times, and very infrequently, central power hungry video processing is performed.

- **SONOPA:** SOcial Networks for Older adults to Promote an Active life [4]
  AAL Joint Programme: 1 May 2013 - 30 April 2016
  Consortium: Docobo Limited, University of Twente, Smart Signs, University of Deusto, SpringTechno, Abotic, E-seniors, Camera-Contact, iMinds/Ghent University, CM.

  Brief description: Sonopa aims to employ a set of available ICT technologies to develop an end-to-end solution for stimulating and supporting activities at home. SONOPA aims to achieve its objective through a data collection and fusion infrastructure which merges real measurements of the users' activities in order to encourage activities with their peers. Reminders and recommendations come through personalized easy-to-use wall displays placed at the user home. SONOPA will employ data analysis techniques to derive a model for the wellness of the user along four dimensions: social, eating, leisure habits and mobility. This model will enable the system to track variations in the daily activities over time in order to detect the right time to provide a recommendation. This allows for timely access to quantitative data from the user and allows the activation of individual and social recommendations. Technologies include: (i) measurement systems that monitor and register the activities of the user at home and with their peers, (ii) behavior modeling and user profiling techniques, delivering a pattern of the users' activities over time by analyzing and summarizing the large sensory data and registered logs; and (iii) a user interface providing personalized recommendations and reminders, encouraging activities to the user. The offered recommendations can be in the form of suggesting individual activities at home, such as preparing meals or social interactions with peers, such as setting up a board game at home.

[4] More details can be found at `http://www.sonopa.eu/`

2

Contour and surface models

2.1 Introduction

In this chapter, we propose an adaptive fitting technique for representing sets of contour pixels and image pixels with geometric primitives. A geometric primitive is a polynomial function describing the geometry of an edge or the variation of grey values in a region. The problems of extracting such primitives arises in various contexts in computer vision: segmentation, approximation and analysis of objects in images.

The problem we study is that of finding a region of maximal size in which grey values can be well approximated by a polynomial function and where contour pixels can be well approximated by polynomials as well. We also consider the related problem of finding the best degree of the polynomial function. Once the segments found, another problem is finding the fitting parameters of the best fit.

To achieve above-mentioned purposes, we propose an adaptive region growing algorithm based on constructive polynomial fitting. Region growing is the process of examining neighbouring pixels of initial seed pixels and determining whether the neighboring pixels should be added to the region. Polynomial fitting fits geometric primitives to pixels. This primitive extraction algorithm determines subsets of pixels that lie on a geometric primitive or close to it. How well a subset corresponds to a primitive is quantified by an L_∞ fitting cost (approximation error). This fitting cost can be computed without computing the best fitting polynomial. The best fit has only to be computed when the grouping of the segment is finished.

In this work, L_∞ fitting costs are computed by constructive fitting. The emphasis of constructive fitting is on the calculation and estimation of the fitting cost from elemental subsets. Elemental subsets are the smallest subsets that have a non-trivial fitting cost. Estimating the fitting cost with elemental subsets boils down to a sampling of the region. Sampling in this context refers to measuring the fitting cost in only a few pixels of the region and then estimating the fitting cost of the entire region based on these measurements. The selection of elemental subsets in a region can be performed locally as well as globally. We define a local elemental subset as an elemental subset which has been selected from the pixels in a small part of the region. Likewise, a global elemental subset is an elemental subset which has been selected from the pixels in the entire region. From local and global elemental subsets we compute so-called local and global fitting costs, respectively. Local and global fitting costs can be combined in several ways. In this work, our key idea is to combine local and global fitting costs with a strategy for contour segmentation and image segmentation in a region growing process with adaptive thresholding. The proposed region growing method examines local fitting costs to decide if a new point is to be added to a segment (to identify edges and outliers), while global fitting costs control if the polynomial degree is adapted. Because the method is adaptive, it bears some resemblance to deformable models [Yuille et al., 1992]. In our case, however, adaptive refers to the use of a local neighbourhood to add pixels, while adapting the shape (or degree) of the function is based on global behaviour. In this sense there is some local flexibility, while the global behaviour is determined by a more straightforward characterization, such as being concave or convex.

We investigate adaptive region growing for contour segmentation as well as for image segmentation. Adaptive region growing for both is based on the same principles of local and global sampling of the region. However, mainly due to the different spatial ordering of contour pixels and image pixels, both need their own modifications of adaptive region growing. Where region growing groups pixels along one dimension for connected contour pixels, region growing groups pixels along two dimensions for connected image pixels. In both cases, this requires different strategies when selecting elemental subsets. Moreover, different polynomial fitting functions for contour regions and image regions imply different computations for constructive fitting.

In this work, the focus is on the segmentation of images and videos of faces. In order to find segments that coincide well with meaningful facial features in the image, we investigate the grouping of pixels into meaningful segments and the approximation of segments by low-degree polynomial functions. The 1-D case firstly segments contour pixels in face images into contour segments. Then, these contour segments are represented by polynomial curves, which are either straight, convex or concave. The 2-D case firstly segments image intensities of face images

into surface segments (intensity patches). Then, these surface segments are represented as polynomial surfaces, which are either flat, planar, convex, concave or saddle surfaces. The polynomial curves and surfaces are grouped into two novel compact features: the Curve Edge Map (CEM) and the Surface Intensity Map (SIM), respectively.

The work in this chapter was published in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

This chapter is structured as follows: in Section 2.2, we discuss related work. In Section 2.3, we explain the basic principles of constructive polynomial fitting. In Section 2.4, we propose an adaptive region growing algorithm based on constructive polynomial fitting. This method is applied and evaluated for (part A) contour segmentation into polynomial curves and (part B) image segmentation into polynomial surfaces in Sections 2.5 and 2.6, respectively.

2.2 Related work

This section gives an overview of existing point, curve and region extraction methods. Feature detection is the extraction of salient structures in images. A feature is any location in the image which indicates a region, line or point of interest. Significant regions, curves (region boundaries) or points (region corners, line intersections, points on curves with high curvature) are considered as features here. They should be distinct, spread all over the image and efficiently detectable in images. Thus, the main purpose of feature extraction is to select features in the image that are likely to be useful candidates for higher-level operations, such as matching, tracking or shape analysis. In practice, areas of interest often correspond to meaningful object parts, in order to describe, recognize or track objects over different images. An overview of the wide variety of feature detectors that exist in literature is given by Schmid [Schmid et al., 2000], Tuytelaars [Tuytelaars and Mikolajczyk, 2008] and Nixon [Nixon and Aguado, 2012]. As in this work, polynomial fitting is one useful way for feature extraction.

Point features. In this paragraph, we give an overview of point extraction methods. Related with the work in this thesis, point features are of interest as input for 1-D polynomial fitting. There are two well-known classes of feature point extraction methods: contour-based and intensity-based. Contour-based methods examine the curvature of edges in the image. Mostly, an edge detector is used as a pre-processing stage to obtain the point set for the desired contour. There is an abundance of edge detectors in use these days, like the Sobel, the Laplacian of Gaussian and the Canny edge detector [Canny, 1986]. A search along the connected edge chains returns meaningful locations, such as points with special characteristics, e.g., highest (change in) curvature, inflexion or intersection points, junctions, endings, etc. Interesting points can also be detected on an approximation

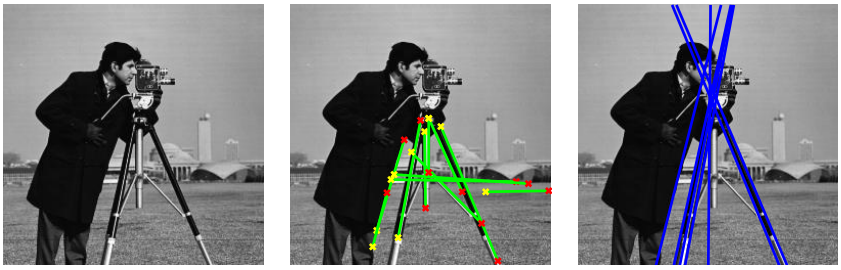
(a) (b) (c)

Figure 2.1: (a): The standard test image *Lena*. (b) and (c): The point feature detection output of the Harris corner detector and the SIFT feature detector, respectively. The SIFT features are plotted with an indication of scale and orientation as indicated vector.

by curves or splines. Intensity-based methods return feature locations based on a measure derived directly from the intensity values. Intensity-based feature point detectors examine small image regions to see whether their appearance mimics that of a corner. These methods check whether the intensity variations in a predefined structure verify a set of similarity rules. Different variations of similar rules are applied in different methods, such as the Moravec detector [Moravec, 1977], the Harris detector [Harris and Stephens, 1988], the SUSAN detector [Smith and Brady, 1997] or the FAST detector [Rosten and Drummond, 2005, Rosten and Drummond, 2006]. The advantage of these types of methods is their computational efficiency. Whereas the above detectors achieve translation and rotation invariance up to some extent, most recent feature detection methods such as SIFT [Lowe, 2004] and SURF [Bay et al., 2006, Bay et al., 2008] consider a scale space approach to obtain (at least) scale invariance. Other detectors explore region-based methods to obtain affine invariance, such as MSER [Matas and Chum, 2004] and ASIFT [Morel and Yu, 2009]. Figures 2.1 (b) and (c) show the point feature detection output of the Harris corner detector and the SIFT feature detector when performed on the standard test image *lena* in Figure 2.1 (a). An evaluation of the performance of several feature descriptors is given by Mikolajczyk [Mikolajczyk et al., 2005, Mikolajczyk and Schmid, 2005].

Existing techniques examine local behaviour on the basis of derivatives or local transitions of grey values, such as in the SUSAN detector or in the FAST detector. Our approach, where fitting is an essential part, is an indirect way to know where the derivatives are zero. The difference is that we work on a local basis as well as on a global basis.

Curve features. Several solutions exist to find geometric shapes in edge data, such as straight lines or conics (circles, ellipses, parabolas). For faces curves cor-



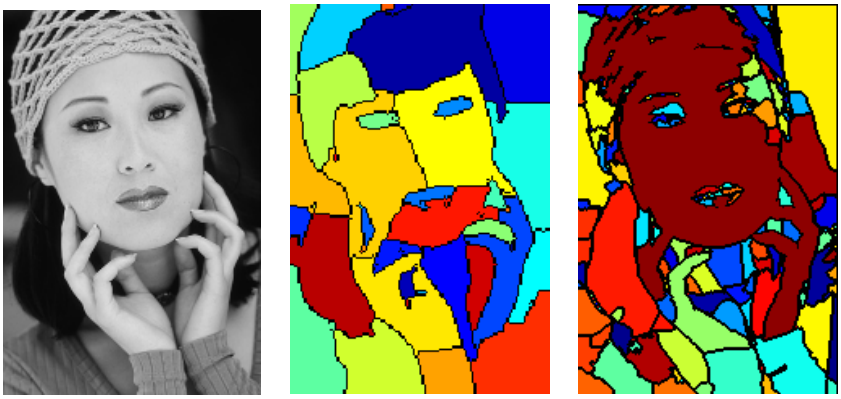
(a) (b) (c)

Figure 2.2: (a): The standard test image *Cameraman*. (b) and (c): The line feature detection output of the Hough transform and the RANSAC algorithm, respectively.

respond to boundaries between segments. Due to imperfections in either the image data or the edge detector, there may be outliers for the desired model, such as spatial deviations from noisy edge points to the geometric model. Therefore, it is often non-trivial to group the extracted edge pixels to an appropriate set of geometric primitives. Common methods to robustly fit a geometric model, here straight lines, to the edge maps are the Hough transform [Hough, 1962, Duda and Hart, 1972, Ballard, 1981] and the RANSAC algorithm [Fischler and Bolles, 1981]. Characteristic for these techniques is their use of parameters of geometric primitives. The Hough transform maps each point of the data set onto a manifold in the parameter space. A minimal subset is the smallest subset that uniquely defines a primitive; e.g., two points define a straight line. A minimal subset based extraction algorithm will repeatedly choose a minimal subset, compute the corresponding primitive parameters, generate the primitive, and verify whether a sufficient number of points in the data set lie sufficiently close to this primitive [Roth and Levine, 1993]. In computer vision, the RANSAC algorithm was one of the first well known techniques based on this principle. Figures 2.2 (b) and (c) show the line feature detection output of the Hough transform and the RANSAC algorithm when performed on the standard test image *cameraman* in Figure 2.2 (a).

In our approach, we use elemental subsets instead of minimal subsets. An elemental subset includes one extra pixel. As such it not only takes into account the parameters of the primitive, but also includes a margin of how far the pixels can be from a primitive.

Region features. Region feature extraction methods fit a parametric intensity model to local image regions. These parametric model methods require image segmentation to divide an image into image regions. Over the past few decades, 2-D image segmentation has been studied extensively with a huge number of algorithms being published in the literature [Fu and Mui, 1981, Haralick and Shapiro,



(a) (b) (c)

Figure 2.3: (a): A face image of the BSDS300. (b) and (c): The region feature output of the normalized cuts algorithm and the mean shift algorithm.

1985, Nevatia, 1986, Pal and Pal, 1993, Muñoz et al., 2003]. Image segmentation approaches are often divided broadly into three categories: *feature-based* [Otsu, 1979, Kapur et al., 1985, Deng and Manjunath, 2001, Comaniciu and Meer, 2002, Chen et al., 2005, Aujol and Chan, 2006, Tao et al., 2007, Mignotte, 2008], *region-based* [Vincent and Soille, 1991, Shih and Cheng, 2005, Makrogiannis et al., 2005, Yang et al., 2008, Tarabalka et al., 2010, Liu et al., 2011b] and *graph-based* [Shi and Malik, 2000, Wang and Siskind, 2003, Couprie et al., 2011]. Feature-based image segmentation collects the main characteristics of an image by extracting image features, which are usually based on colour or texture. The feature samples are represented as vectors. The objective is to group the extracted feature vectors into well-separated clusters by using a specific distance metric. Drawbacks of these methods are the non-preservation of spatial structure and edge information and the possible grouping of pixels from disconnected regions of the image with overlapping feature spaces. In the spatial domain, region-based image segmentation preserves edge information and spatial relationship between pixels in an image. The objective is to detect regions that satisfy predefined criteria in a region-growing, region-merging or region-splitting process. An example of a region-based image segmentation criterium is based on polynomial fitting. Graph-based image segmentation groups the feature-based and region-based information. Grouping is based on several important elements, such as similarity, proximity, and continuation. A weighted graph can be constructed, where each vertex corresponds to a pixel or region, and the associated weight of each edge connecting two adjacent pixels or regions depends on the likelihood that they are belonging to the same region. The weights are often related to colour and texture features.

Figures 2.3 (b) and (c) show the region feature detection output of the normalized cuts algorithm [Shi and Malik, 2000] and the mean shift algorithm [Comaniciu and Meer, 2002] when performed on an image of the BSDS300 [Martin et al., 2001] in Figure 2.3 (a). In our method, grouping is mainly based on a rigorous form of similarity and continuity.

As in this work, segmentation of intensity images with polynomial fitting is originating from Segmented Image Coding (SIC) [Eden and Kocher, 1985, Kunt et al., 1987, Gilge et al., 1989, Philips, 1996, Christopoulos et al., 1997, Reid et al., 1997, Biswas, 2003, Kassim et al., 2009]. The main idea of SIC is to divide the image into segments that coincide as well as possible with meaningful parts in the image. Each region is represented by two codes. The first (often a chain code) describes the location of contour pixels. The second represents the best approximation of the region enclosed by this boundary. For these techniques, the main purpose was image compression.

In literature, researchers have mainly been investigating surface segmentation in range images [Fan et al., 1989, Lim et al., 1990, Taubin, 1991, Wang et al., 2012]. One popular approach for surface segmentation is region growing, for which several segmentation criteria have been proposed. A distinction is made between segmentation based on normal vectors [Rabbani et al., 2006, Deschaud and Goulette, 2006], curvatures [Besl and Jain, 1988, Vieira and Shimada, 2005, Lavoué et al., 2005, Jagannathan and Miller, 2007, Wang and Yu, 2011] and fitting polynomials [Kocher and Leonardi, 1986, Wang et al., 2003, Cohen-Steiner et al., 2004]. Existing techniques using low-degree polynomial fitting for surface segmentation consider planar patches [Cohen-Steiner et al., 2004] and quadratic patches [Wang, 2002, Petitjean, 2002, Yan et al., 2012], such as circular cylinders [Wu and Kobbelt, 2005] and ellipsoidal surfaces [Simari and Singh, 2005]. These techniques are often not suited to segment intensity images. This is caused by the different behaviour of range images and intensity images. Range images are often smooth and represent the real object, while grey values in intensity images are more textured because they represent reflected light. Therefore, segmenting intensity images with region growing [Zucker, 1976, Adams and Bischof, 1994, Kanga et al., 2012] requires an adaptive approach which can handle the local and global variation of grey values [Pohle and Toennies, 2001, Qin and Clausi, 2010].

2.3 Constructive polynomial fitting

In this section, we discuss the main aspects of constructive polynomial fitting. Constructive fitting is a technique for curve and surface fitting, and the related problem of the extraction of geometric primitives from images [Veelaert, 1997, Veelaert and Teelen, 2006, Veelaert, 2012]. A geometric low-level feature or primitive is a polynomial function describing the geometry of an edge or the variation

of grey values in a region. Primitive extraction arises in various contexts in computer vision: segmentation, approximation and analysis of objects in images [Roth and Levine, 1993, Taubin, 1991, Bolle et al., 1992]. A primitive extraction algorithm finds subsets of pixels that lie on a geometric primitive or close to it. How well a subset corresponds to a primitive is measured by a fitting cost. Thus the extraction algorithm essentially determines subsets with low fitting cost.

In its most general form the problem can be described as follows: find a subset D of a given data set S such that a geometric primitive can be fitted to the pixels of D with a fitting cost below a given threshold. Depending on the context, additional constraints may be imposed on the subset D, such as either being connected, convex, have a minimal or maximal size, or it must satisfy certain distance constraints.

A primitive extraction process based on merging tries to find a geometric primitive in a large data set by first extracting parts or segments of primitives from small subsets of the data set, and then combining them into larger parts. To formalize this process, it is possible to derive from the parameters and fitting costs of the small parts the parameters and fitting costs for the combined parts [Veelaert, 1997]. This is called constructive fitting. The purpose of constructive fitting can be formulated as generally as the primitive extraction problem. From a small subset D that has a lowest fitting cost, we try to construct a larger subset with low fitting cost.
Constructive fitting involves the following principles:

- Constructive fitting is based upon uniform fitting (also called minimax, Chebyshev or L_∞ fitting).
- The smallest subsets that have a nontrivial fitting cost will be called *elemental subsets* [Stromberg, 1993]. The fitting cost of an elemental subset can be computed by a simple formula.
- The fitting cost of a data set S is equal to the maximum of the fitting costs of all its elemental subsets.
- The fitting cost of a data set S can be *estimated* from the fitting costs of a few of its elemental subsets provided the elemental subsets form a so-called *rigid* collection [Veelaert, 1997].
- Rigid collections of elemental subsets can be built as follows: construct an ordered sequence of elemental subsets that cover S such that the intersection of each subsequent pair in the sequence is a minimal subset [Veelaert, 1997].
- The fitting parameters of the best fit are equal to the fitting parameters of the elemental subset that has the largest fitting cost [Veelaert, 1997].

Figure 2.4 illustrates linear polynomial fitting for a data set S whose points have been marked by dots. The fitting cost can be found by estimating a line

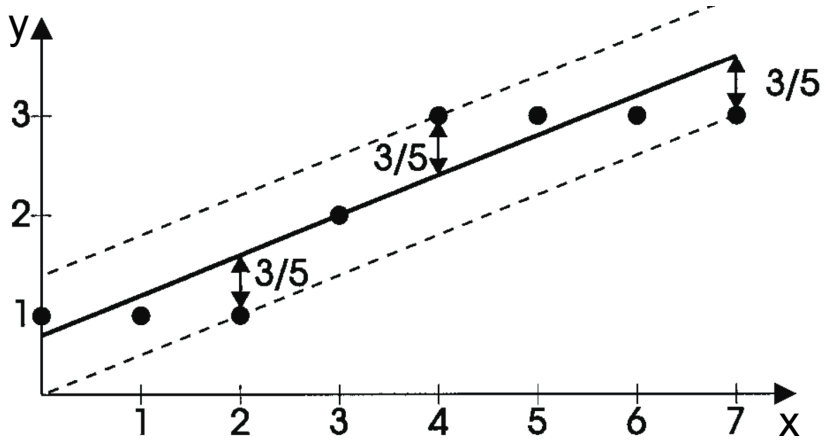


Figure 2.4: Best linear fit according to the L_∞ metric. Parallel enclosing lines are shown as dashed lines.

$y = \alpha_1 + \alpha_2 x$ until $\max_{x,y}(|\alpha_1 + \alpha_2 x - y|)$ is minimal. Figure 2.4 also shows the resulting best fitting line, with fitting cost $3/5$. The best fit has a simple geometric interpretation. It can be found by letting two parallel lines move towards each other until they enclose the data set points as tightly as possible; that is, until the vertical distance between lines is as small as possible. The best fit is the line that bisects the region between the enclosing lines, as is also shown in Figure 2.4.

The focus of constructive fitting is on the calculation and estimation of the fitting cost. The fitting parameters are only computed when needed explicitly (after we have determined a subset D whose fitting cost we are interested in), because it is possible to estimate the fitting cost without calculating the parameters. This is in clear contrast with the Hough Transform or the minimal subset approach. A typical constructive fitting process starts from small parts of geometric primitives and assembles them into larger parts. The calculation of the fitting costs of the large parts is avoided whenever possible. Instead their fitting costs are estimated from the costs of the smaller parts. Eventually, when a large part of a primitive has been found with a sufficiently low estimated fitting cost, its fitting parameters are estimated, and if necessary, the parameters and the fitting cost are computed exactly. In this way, constructive fitting is well suited for exploring many possible ways to fit models to the data.

When assembling small parts of primitives it is essential that we start from uniform fits, i.e. L_∞ fits. In contrast with least-squares fitting, a uniform fit retains some information about the actual positions of the pixels. If we uniformly fit a primitive to a given set of pixels, we also have precise information about the region

in which the pixels lie. Furthermore, there is always an elemental subset that has the same uniform best fit as the entire data set [Veelaert, 1997]. This property ensures that the elemental fits can be used as building blocks for larger fits.

Each time part of a primitive has been extracted, there is always one elemental subset that holds sufficient information about the positions of the pixels of the part. Sufficient information here means that we can use the elemental subset to estimate the resulting fitting cost when we start merging it with other elemental subsets. As mentioned by Stromberg, the importance of elemental subsets in uniform fitting was already known for linear regression [Stromberg, 1993]. We use elemental subsets to more general fitting problems, using a consequence of one of Helly's theorems on convex sets [Stoer and Witzgall, 1970, Kelley and Weiss, 1979, Veelaert, 1993, Veelaert, 1994].

Apart from its suitability for merging, constructive fitting based on the L_∞ fitting cost has some advantages over the L_2 fitting cost [Watson, 1998, Cadzow, 2002], which makes it also suitable as a stopping criterion for region growing and to detect outliers and edges. We demonstrate this with a small example. Suppose we use a region growing process to fit lines $g(x) = ax + b$ to segments $S = \{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_m, f(x_m))\}$ of a noisy step function $f(x)$, as in Figure 2.5 (a). The first point of the step function at the left is taken as a seed point. The corresponding L_∞ and L_2 fitting costs are shown in Figure 2.5 (b). The L_2 fitting cost of $g(x)$ over a segment S is defined as

$$r_{L_2}(S; g) = \sqrt{\sum_{x \in S} (g(x) - f(x))^2}. \tag{2.1}$$

The L_∞ fitting cost of $g(x)$ over a segment S is defined as

$$r_{L_\infty}(S; g) = \max_{x \in S} |g(x) - f(x)|. \tag{2.2}$$

The step function causes a much faster and more direct increase of the consecutive L_∞ fitting costs than the consecutive L_2 fitting costs. Moreover, once the step occurs, the L_2 costs gradually decrease. The obvious reason is that L_2 takes an average over all deviations between the pixels and the fitting line, while L_∞ looks at the maximum deviation. Consequently, when making decisions during region growing about adding a new pixel, discarding an outlier or stopping at an edge, the direct response of the L_∞ fitting costs yields more sensitivity as well as accuracy.

2.4 Adaptive region growing

In this work, we propose a novel technique to group pixels into smooth segments with adaptive region growing. Before going into detail in the region growing processes for contour segmentation into polynomial curves in Section 2.5 and for

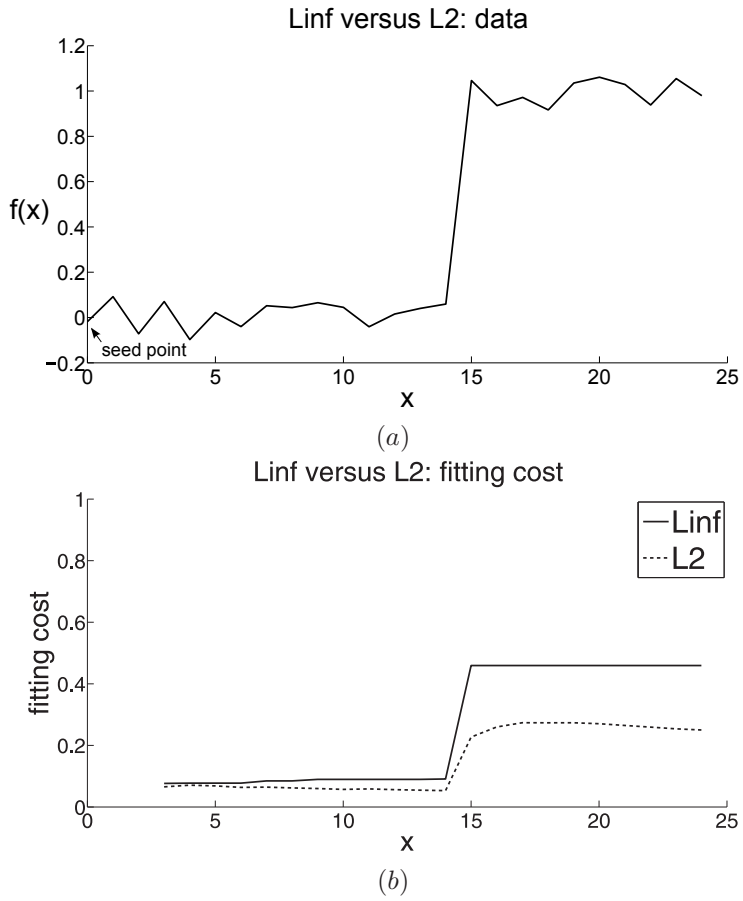


Figure 2.5: (a): A noisy step function. (b): The corresponding L_∞ and L_2 fitting costs.

image segmentation into polynomial surfaces in Section 2.6, we highlight the concepts of the proposed region growing techniques which are similar for both contour segmentation and image segmentation.

The purpose of region growing is to group contour pixels and image pixels into segments that satisfy a certain criterion, such as being straight or smooth. To find such segments, a typical region growing algorithm starts from a small seed segment, and then repeatedly tries to add new pixels to this segment, while verifying whether the segmentation criterion is still satisfied for the enlarged segment. If not, a new segment is started, or another pixel is chosen. For region growing, we propose adaptive thresholding of the L_∞ fitting cost as a stopping criterion

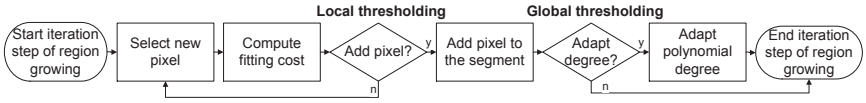


Figure 2.6: Flow chart of one iteration step in our adaptive region growing. Two key phases are distinguished: the first determines if a new pixel is to be added to a segment, while the second phase controls if the polynomial degree is adapted.

and to distinguish outliers and edges from gently rising variation. An outlier is a grey value differing from its neighbours due to noise or small object speckles. The novelty is that region growing investigates the local variation of grey values in a segment to identify edges and outliers, while the global variation of grey values in a segment is investigated to adapt the degree of the polynomial function. The terms local and global refer to a part of the segment and the entire segment, respectively. The combination of both is possible because we employ constructive fitting: the global fitting cost is also calculated from local fitting costs, which is more economical. The greatest advantage, however, is that there is a clear relation between local and global fitting costs. As we will show, the global fitting cost is computed by evaluating all possible elemental subsets while the local fitting cost is computed from a small subcollection. Adaptive thresholding allows for a variable polynomial degree and a variable fitting error, depending on the local properties of the pixels. We find regions of maximal size that satisfy an L_∞ fitting cost criterion, such that grey values can be approximated well by polynomial functions of low degree (e.g., 0, 1 or 2).

In this work, region growing is made adaptive to accommodate the smooth variation of the grey values over a segment. Two key phases are distinguished, which are visualised in the flowchart in Figure 2.6. The first phase determines if a new pixel is to be added to a segment. The second phase controls if the polynomial degree is adapted. These phases are driven by adaptive thresholding of the L_∞ fitting cost. In the following sections, we explain adaptive region growing with constructive polynomial fitting for 1-D contour as well as for 2-D image segmentation. The methods in both cases are based on the same principles of local and global sampling of the region. However, mainly due to the different spatial ordering of contour pixels and image pixels, both need their own modifications of adaptive region growing. Where region growing groups pixels along one dimension for connected contour pixels, region growing groups pixels along two dimensions for connected image pixels. In both cases, this requires different strategies when selecting elemental subsets. Moreover, different polynomial fitting functions for contour regions and image regions imply different computations for constructive fitting.

2.5 Contour segmentation into polynomial curves

In this section, we propose a simple, linear-time algorithm for 1-D segmentation of digitized contours into polynomials of variable degree, which we illustrate for linear and parabolic segments. The main purpose is to show how local and global fitting costs can interact in an adaptive method. Linear time complexity, simplicity and generality do not come for free, however. The method is based on the estimation of the fitting error, not the exact computation, and therefore the segmentation will only approximately estimate the criterion. The deviation from optimality can be reduced, however, by increasing the number of samples, and for most applications the error probability can be made sufficiently small. This work was published in [Deboeverie et al., 2010].

2.5.1 Contour extraction

Mostly, in contour segmentation an edge detector is used to detect edges in a preprocessing step to obtain the pixel set for the desired contour. There is an abundance of edge detectors in use these days, like the Sobel, the Laplacian of Gaussian and the Canny edge detector [Canny, 1986]. Beside an edge detector, contours can also be obtained from the contours separating image segments.

In our work, the edge map is derived from the Canny edge detector. The Canny edge detector is suitable for us, because it results in thin edges of one pixel thickness and it is less sensitive to noise. Connected contours are obtained from the edge map by a simple boundary scan algorithm. A typical example of the output of an edge detector is given in Figure 2.7.

2.5.2 Constructive polynomial curve fitting

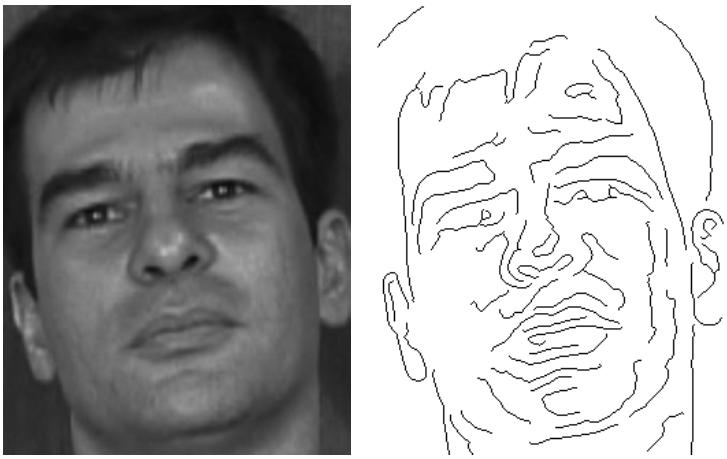
In this section we give the mathematical formulation of constructive polynomial curve fitting on which the region growing processes in this work are based.

Let $p_i = (x_i, y_i) \in \mathbb{Z}^2$ be the pixels of a finite contour $C_k = \{p_0, \ldots, p_k\}$. Let G be a vector space of fitting functions, for instance, the $(d+1)$-dimensional vector space of polynomial curves of the form

$$g(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_d x^d. \tag{2.3}$$

To simplify the properties that follow, we impose the mild constraint that the segment $C_m = \{p_0, \ldots, p_m\}$, which is part of C_k, contains at least $d+1$ distinct pixels, $d+1 \leq m \leq k$, where d denotes the dimension of the vector space of the fitting functions G or d denotes the polynomial degree of the fitting functions. The cost of uniformly fitting $g(x)$ to the contour C_m is defined as

$$r_d(C_m) = \max_{(x_i,y_i)\in C_m} |g(x_i) - y_i|. \tag{2.4}$$



(a) (b)

Figure 2.7: (a): A greyscale image of a face. (b): The result after Canny edge detection.

Note that this is also called the Chebyshev, minimax, or L_∞ fitting cost. The best fit is the function $g(x)$ in G for which $r_d(C_m)$ is minimal. We denote this minimal cost as $\hat{r}_d(C_m)$, and we call it the fitting cost over C_m. To be precise,

$$\hat{r}_d(C_m) = \min_{g \in G} r_d(C_m). \tag{2.5}$$

The best fit is unique if the pixels are in general position. The solution is not unique if the pixels are in an algebraic relation, such as collinearity. The first property of constructive fitting that we need is that the best fit and its fitting cost can be computed from fits to the so-called elemental subsets of C_m. These are subsets of the curve C_m that contain precisely $d+2$ pixels and have a nontrivial fitting cost. The fitting cost over an elemental subset itself can be computed in a straightforward manner. To be precise, let $D = \{(x_1, y_1), \ldots, (x_{d+2}, y_{d+2})\}$ be an elemental subset of C_m. Let E_j denote the cofactor of y_j of the augmented matrix:

$$(A_D|B_D) = \left(\begin{array}{cccc|c} 1 & x_1 & \ldots & x_1^d & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{d+2} & \ldots & x_{d+2}^n & y_{d+2}) \end{array}\right) \tag{2.6}$$

Then one can show [Veelaert, 1997, Veelaert and Teelen, 2006] that the fitting

cost of an elemental subset D can be computed by

$$\begin{aligned}\hat{r}_d(D) &= \frac{\det(A_D|B_D)}{(|E_1| + \ldots + |E_{d+2}|)} \\ &= \frac{(|E_1 y_1 + \ldots + E_{d+2} y_{d+2}|)}{(|E_1| + \ldots + |E_{d+2}|)},\end{aligned} \tag{2.7}$$

provided the denominator is non-vanishing. The denominator is vanishing when the pixels in an elemental subset are not in general position, but in an algebraic relation, such as collinearity. One can prove that the fitting cost over C_m is the maximal value of the elemental fitting costs over all elemental subsets of the contour C_m [Veelaert, 1997, Veelaert and Teelen, 2006].

$$\hat{r}_d(C_m) = \max_{D \in M} \hat{r}_d(D), \tag{2.8}$$

where M is the collection of all elemental subsets D of C_m for which $|E_1| + \ldots + |E_{d+2}| > 0$. We can obtain a reliable estimate of the fitting cost with far fewer computations than required for computing the fitting cost itself [Veelaert, 1997, Veelaert and Teelen, 2006]. The fitting cost of a data set can be estimated from the fitting costs of a few of its elemental subsets. Instead of calculating $\hat{r}_d(C_m)$, we compute

$$\tilde{r}_d(C_m) = \max_{D \in \tilde{M}} \hat{r}_d(D), \tag{2.9}$$

where $\tilde{M}$ forms a rigid subcollection of elemental subsets of M [Veelaert, 1997]. The rigidity of a collection of elemental subsets is related to the rigidity of mechanical constructions [Veelaert, 1997]. To be rigid it is necessary that each pixel is covered by at least one elemental subset, and that each elemental subset, which has m pixels, has at least $m-1$ pixels in common with all of the other sets. (Henneberg sequences [Graver, 1991]). In a region growing process these conditions are met automatically. One can prove that $\tilde{r}_d(C_m) \leq \hat{r}_d(C_m) \leq \gamma \tilde{r}_d(C_m)$, for some value γ that only depends on the way in which the elemental subsets are chosen, not on the pixels themselves [Veelaert, 1997]. γ is the maximal estimation error and can be calculated by solving a mathematical programming problem with constraints that depend on the elemental subset collection $\tilde{M}$. The best way to select elemental subsets is related to the extrema of Chebyshev polynomials (minimax property of de la Vallée-Poussin) [Watson, 2000]. As an example, Figure 2.8 shows a subcollection of 4 elemental subsets, indicated by crosses in different colors, selected on a set of contour pixels. For polynomial curves of degree $d = 2$, the number of pixels in an elemental subset is $d + 2 = 4$. On the left is shown the best fit polynomial curve.

The property that the fitting cost $\hat{r}_d(C_m)$ is equal to the maximum of the elemental fitting costs, as expressed in Eq. (2.8), is of primordial importance for our method. It means that the fitting cost can only increase when we add more pixels

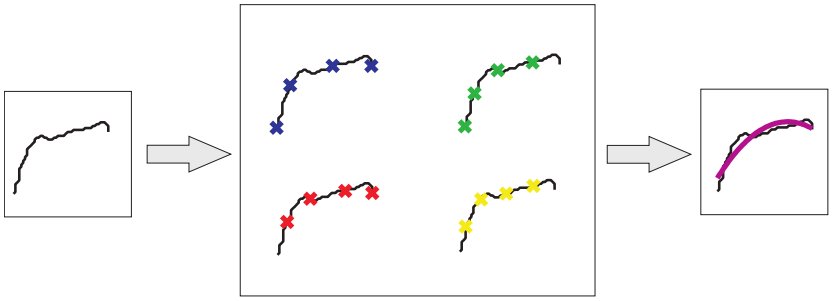


Figure 2.8: Left: a set of contour pixels. Middle: a subcollection of 4 elemental subsets, indicated by crosses in different colors. Right: The best fit polynomial curve.

to a region, which is not always the case for L_2 fitting. Furthermore, if the new pixel falls within the current margin, $\tilde{r}_d(C_m)$ will not increase. On the other hand if the new pixel does not fit well, we will notice a sudden increase in the cost. This property is ideal for segmentation. Suppose that a large set S of pixels contains several segments and a number of outliers. Each elemental subset E corresponds to a random sample of S. By comparing the fitting cost of E with a threshold, we will know whether 1) all pixels in E belong to a common segment 2) the pixels in E belong to distinct segments or E contains at least one outlier. Although we cannot distinguish between elemental subsets that belong to multiple segments and elemental subsets that contain outliers, this property is still very suitable for region growing. If we have a small seed of a segment, we can replace one of the pixels in an elemental subset by a new pixel, and see whether the new pixel is either part of a new segment or an outlier. A second difficulty is that we do not know the shape beforehand. A contour may be straight or curved, a surface patch may be flat, concave or convex. Fortunately, the models that we use are not independent. By increasing the degree of a polynomial we can alter its shape. This is done by comparing a global fitting cost with a threshold.

2.5.3 Contour segmentation with adaptive region growing

In this section, we describe the L_∞ based adaptive segmentation method. We explain how region growing works and how it is made adaptive to the local contour properties, such that contour segments are represented as polynomial curves with a variable polynomial degree and a variable fitting error.

We give an overview of the reasons to introduce an adaptive region growing algorithm instead of region growing by simple thresholding the fitting cost. Two main drawbacks occur by region growing with simple fixed thresholding.

Firstly, due to a fixed threshold on the fitting cost in non-adaptive region growing, the maximum deviation between the contour segments and the polynomial curves is fixed. In this case, we detect endpoints of the segments for instance in the middle of a contour. Figure 2.9(b) shows the first degree polynomial fitting costs $\tilde{r}_1(C_m)$, as defined in Eq. (2.9), when running through the contour from the red cross p_1 to the blue plus p_3, via the green circle p_2, as shown in Figure 2.9(a). A fixed threshold on the fitting cost incorrectly segments the contour in p_3, which corresponds to the blue dotted lines. Instead, we prefer a segmentation of the edge in p_2, which corresponds to the green dotted lines. In adaptive region growing, we propose to detect endpoints of the segments at locations where the direction or the regularity of the contours changes (e.g. at corners). Such points are of interest, because they can be used ase feature points in for instance correspondence problems and object recognition. These changes in direction and regularity correspond to discontinuities in fitting costs.

Secondly, the polynomial curves have a fixed polynomial degree in non-adaptive region growing. When using a fixed polynomial degree under- or even overfitting could occur. First degree polynomials underfit curved contours, while higher degree polynomials overfit straight contours. Figure 2.10 (a),(b) and (c) show the segmentation results for a car image from the MPEG7 database into polynomial functions of degree 1 ,2 and 3, respectively. The polynomials are represented with red dotted lines, green dashed lines and blue solid lines, respectively. The polynomial functions of degree $d = 1$ underestimate the curved contours of the wheels of the car, while the polynomial functions of degree $d = 3$ overestimate the straight contours at the back of the car. In adaptive region growing, we propose to determine the polynomial degree by observing the regularity and the increase of fitting costs. Variable degrees for polynomial functions can potentially avoid under- and overfitting: straight contours are expected to be approximated by lines, while higher degree polynomials are used for curved contours.

In this work, we propose a solution for these problems by making the segmentation of the contours and the decision of the polynomial degree adaptive. We extend constructive polynomial fitting by allowing a variable polynomial degree and variable maximum deviation between the contour segments and the polynomial curves.

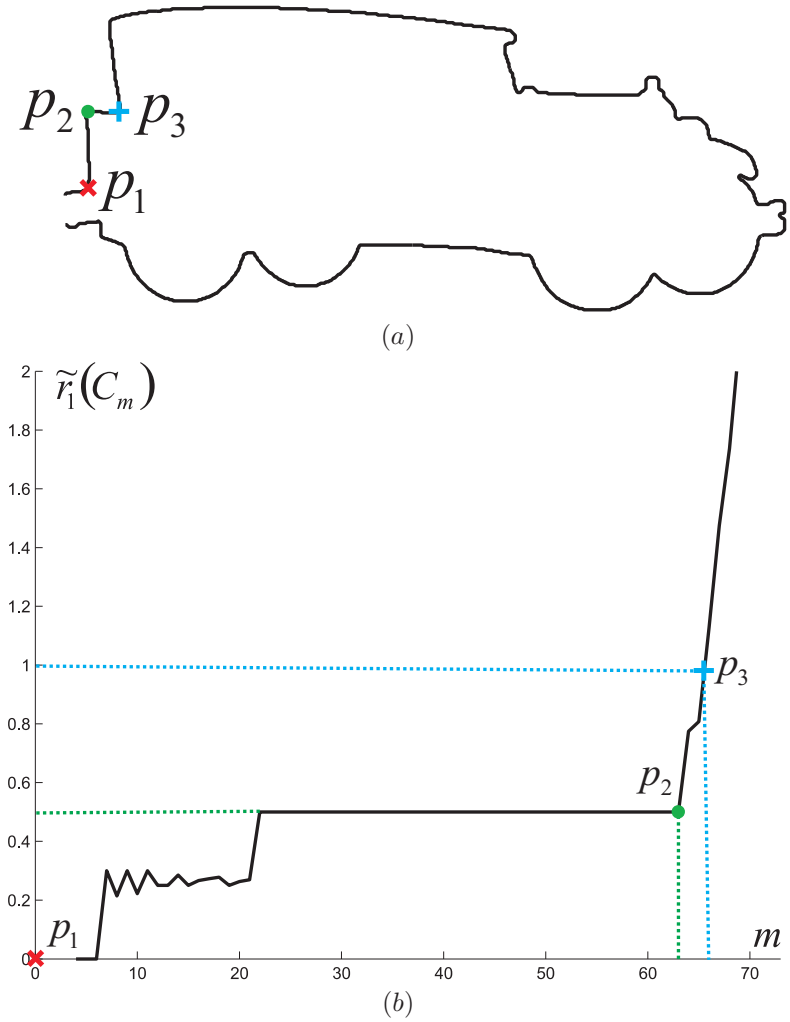


Figure 2.9: (a): The contour of a car image obtained from the canny edge detector. Indication of a run through the contour from the red cross p_1 to the blue plus p_3, via the green circle p_2. (b): The first degree polynomial fitting costs $\tilde{r}_1(C_m)$, as defined in Eq. (2.9). The blue plus p_3 indicates the location of segmentation after thresholding. The green circle p_2 indicates the preferred location of segmentation.

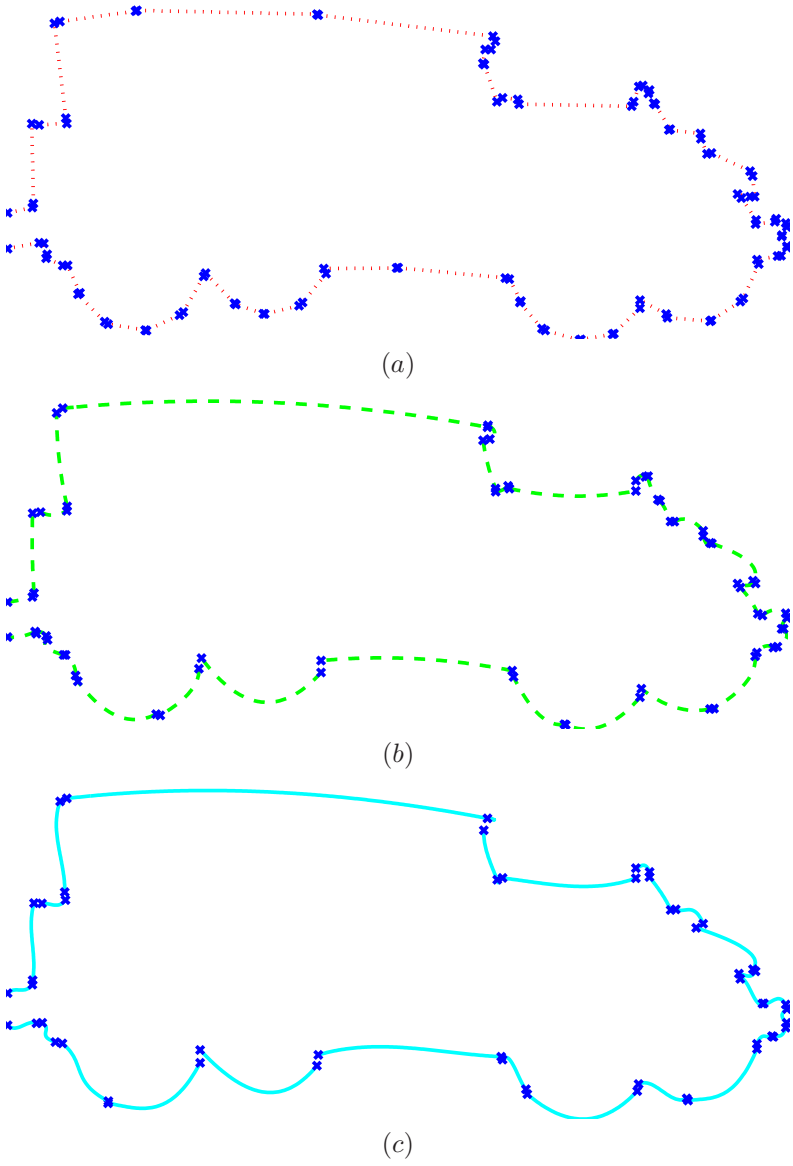


Figure 2.10: Figures (a),(b) and (c) show the segmentation results into polynomial functions of degree 1 ,2 and 3, respectively. The polynomial are represented with red dotted lines, green dashed lines and blue solid lines, respectively. The polynomial functions of degree $d = 1$ underestimate the curved contours of the wheels of the car, while the polynomial functions of degree $d = 3$ overestimate the straight contours from the back of the car.

The segmentation algorithm starts its run through the contour map in three consecutive pixels of the digitized contour, at a randomly chosen starting pixel. At first we attempt to fit first degree polynomial functions, so the initial parameter values are $d = 1$ and $C_2 = \{p_0, p_1, p_2\}$. When extending the curve C_m with a new pixel, $m = m + 1$, we evaluate the value and the regularity of the consecutive fitting costs $\tilde{r}_d(C_m)$ (Eq. 2.9), in order to make decisions about segmentation and polynomial degree.

The segmentation is made dependent on the regularity of consecutive fitting costs. More precisely, we segment contours in locations where the direction or the regularity of the contours change, which correspond to discontinuities in the consecutive fitting costs. To avoid discontinuities in the fitting costs due to noise, we perform mean filtering with a sliding window.

The current moving average $A_d^u(C_m)$ of the fitting costs is

$$A_d^u(C_m) = \frac{1}{|u|} \sum_{i=1}^{u} \tilde{r}_d(C_{m-u+i}), \tag{2.10}$$

where u is the window size of the current moving average, e.g. $u = 5$. To detect discontinuities in the fitting costs, we also observe the future moving average

$$A_d^v(C_m) = \frac{1}{|v|} \sum_{i=1}^{v} \tilde{r}_d(C_{m+i}), \tag{2.11}$$

where v is the window size of the future moving average, e.g. $v = 5$. Discontinuities in the fitting costs are detected in the evaluation of the value I_1, which we define as the difference in future moving average $A_d^v(C_m)$ and current moving average $A_d^u(C_m)$,

$$I_1 = A_d^v(C_m) - A_d^u(C_m). \tag{2.12}$$

A new segment is found, if the value I_1 exceeds the threshold T_1, e.g. $T_1 = 0.6$. The process starts a new segment with the remaining pixels of the contour map for polynomial functions with degree $d = 1$. Figure 2.11 (a) indicates a run through the contour from the red cross to the green circle, for which the fitting costs $\tilde{r}_1(C_m)$ are plot in Figure 2.11 (b). The magenta dashed line, the orange dotted line and the blue dashed-dottted line correspond to the current moving average $A_1^u(C_m)$, the future moving average $A_1^v(C_m)$ and the value I_1, respectively.

The degree of the polynomial function is determined by observing the value I_2, which is defined as the current moving average $A_d^u(C_m)$,

$$I_2 = A_d^u(C_m). \tag{2.13}$$

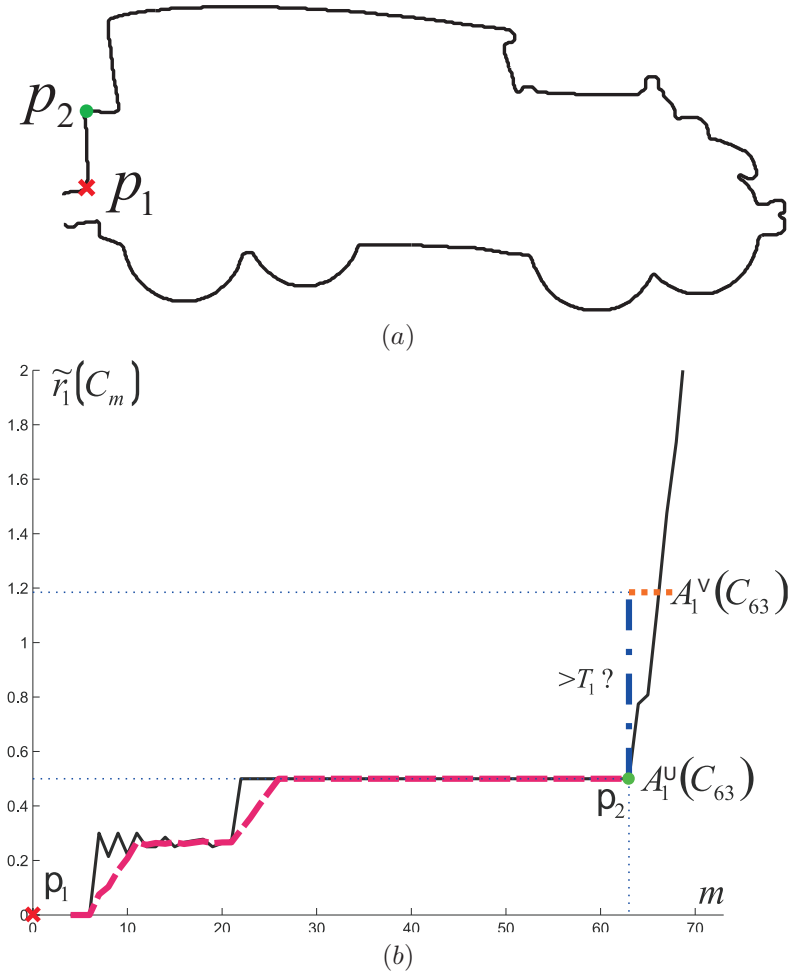


Figure 2.11: (a): A run through the contour from the red cross p_1 to the green circle p_2. (b): The corresponding first degree fitting costs $\tilde{r}_1(C_m)$. The magenta dashed line, the orange dotted line and the blue dashed-dottted line correspond to the current moving average $A_1^u(C_m)$, the future moving average $A_1^v(C_m)$ and the value I_1, respectively.

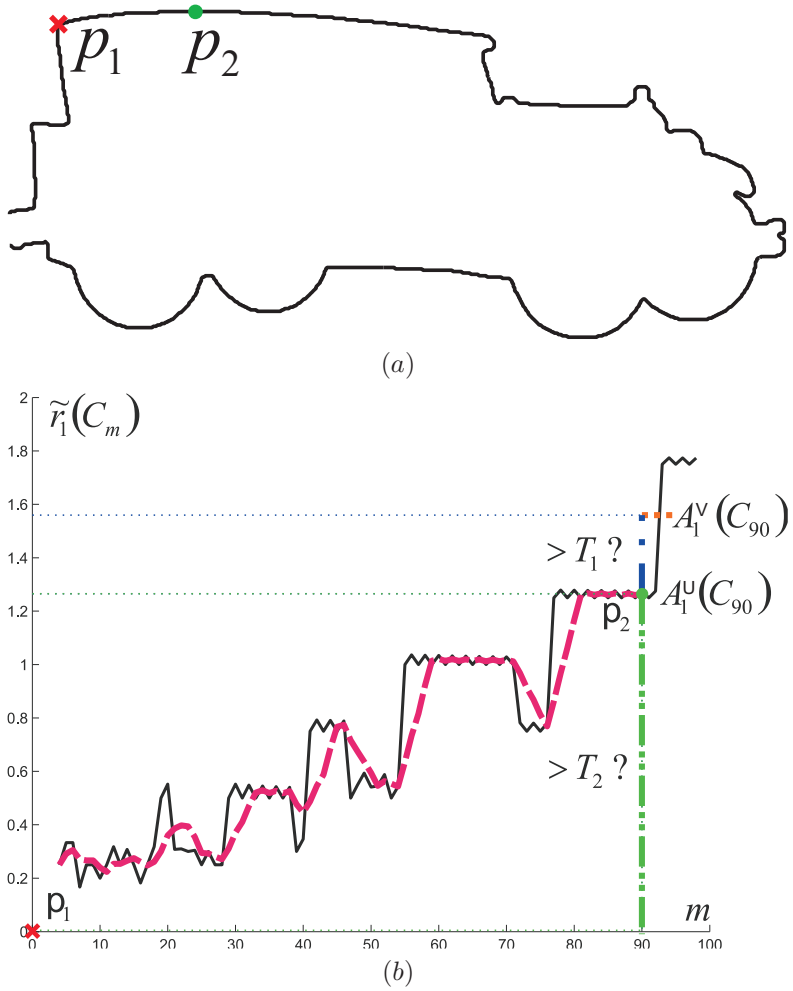


Figure 2.12: (a): A run through the contour from p_1 to p_2. (b): The corresponding first degree fitting costs $\tilde{r}_1(C_m)$. The same notations, markers and colors are used as in Figure 2.11. The value I_2 corresponds to the green dashed-double dotted line.

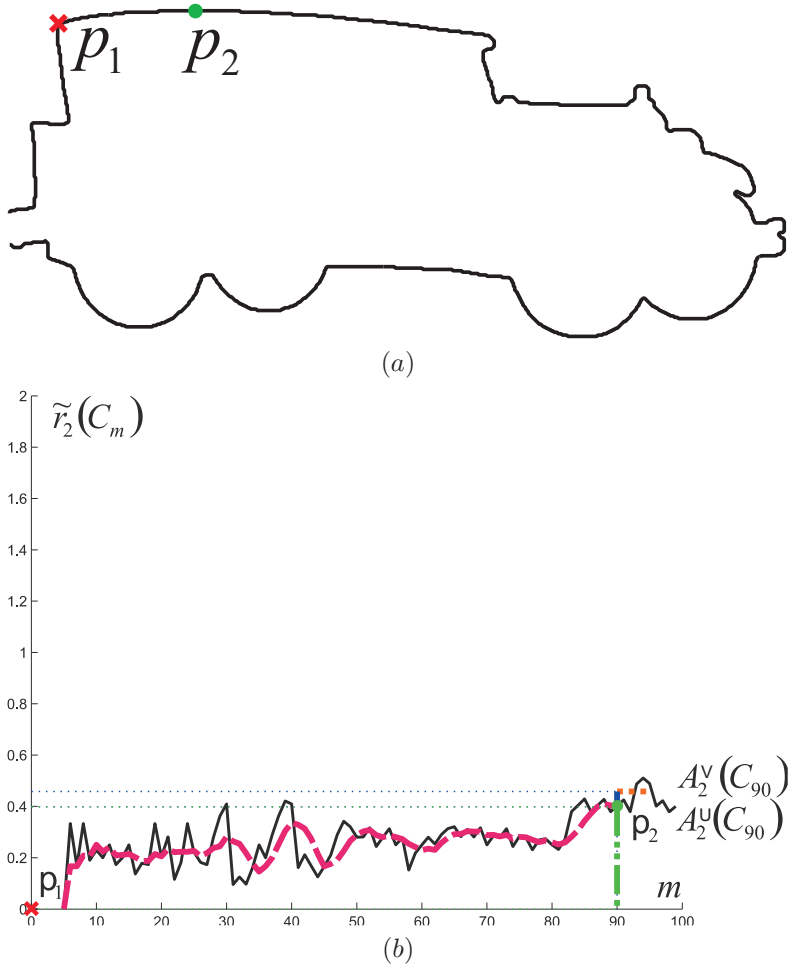


Figure 2.13: (a): A run through the contour from the red cross p_1 to the green circle p_2. (b): The corresponding second degree fitting costs $\tilde{r}_2(C_m)$. The same markers and colors are used as in Figure 2.11. The notations for the current moving average and future moving average are $A_2^c(C_m)$ and $A_2^f(C_m)$, respectively. The value I_2 is again within the limit of T_2.

(a) (b) (c)

Figure 2.14: (a): A greyscale image of a face. (b): The result after Canny edge detection. (c): The face contours segmented into polynomial curves of second degree.

If the value I_2 exceeds the threshold T_2, e.g. $T_2 = 1.3$, the degree of the polynomial function is increased by one, $d = d + 1$. The degree is increased until the value I_2 is again within the limit of the threshold T_2. The segmentation process continues with the remaining pixels of the digitized contour. Figure 2.12 (a) indicates a run through the contour from the red cross p_1 to the green circle p_2, for which the fitting costs $\tilde{r}_1(C_m)$ are plot in Figure 2.12 (b). The green dashed-double dotted line corresponds to the value I_2. When the degree is increased, as shown in Figure 2.13, the value I_2 is again within the limit of T_2.

For each extension of the segment C_m with a new pixel, we evaluate if I_1 and I_2 exceed T_1 and T_2, respectively. The first value is responsible for a segmentation of the contour, while the second value is responsible for an increase of the polynomial degree.

2.5.4 Best fit polynomial curve - Curve Edge Map

After segmentation the coefficients of polynomial curves are optimized by polynomial regression. Recall that during region growing, only the fitting cost was computed. The output of the fitting algorithm is a list of polynomial curves that approximate the contour map. We introduce a compact feature, the Curve Edge Map (CEM), which integrates structural information and spatial information by grouping pixels of an contour map into contour segments. Between 20 and 50 polynomial curves are sufficient to describe a face.

As illustrated in Figure 2.14 (c), many polynomial curves in a face CEM correspond to physically meaningful features such as eyebrows, cheekbones or lips. The polynomial curves approximate the contours very closely. CEM is a simple

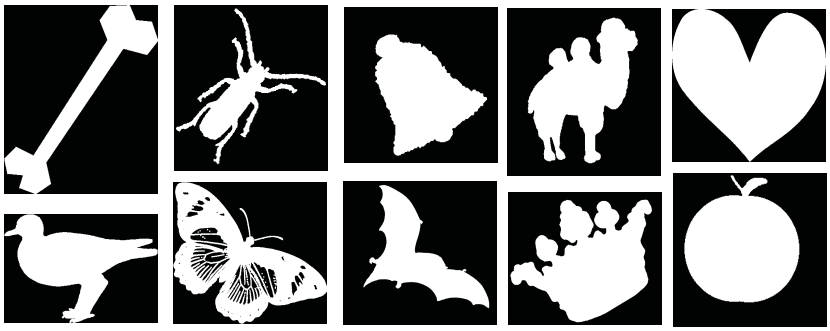


Figure 2.15: Examples of the MPEG-7 core experiment CE-Shape-1 database part B [Jeannin and Bober, 1999].

and natural description which still preserves sufficient information about the facial position and expression. CEM is expected to be less sensitive to illumination changes, because they are intermediate-level image representations derived from a low-level contour map representation.

2.5.5 Evaluation of contour segmentation and approximation

We evaluate the proposed contour segmentation and approximation on the MPEG-7 core experiment CE-Shape-1 database part B [Jeannin and Bober, 1999] containing binary images of shapes with single closed contours. A few examples are shown in Figure 2.15.

We denote segmentation with region growing by thresholding as Non-Adaptive Constructive Polynomial Fitting (NACPF) and the proposed segmentation with region growing by adaptive thresholding as Adaptive Constructive Polynomial Fitting (ACPF).

The values for the parameters of ACPF in Section 2.5.3 are $u = 5$, $v = 5$, $T_1 = 0.6$ and $T_2 = 1.3$. These parameters have been manually tuned on a small number of images. The system is implemented in C++ as an application running on a 2.80 GHz processor, 4.00 GB RAM and 64-bit operating system. Considering an image size of 200x200, the CEM computation time is approximately one 35 milliseconds per image.

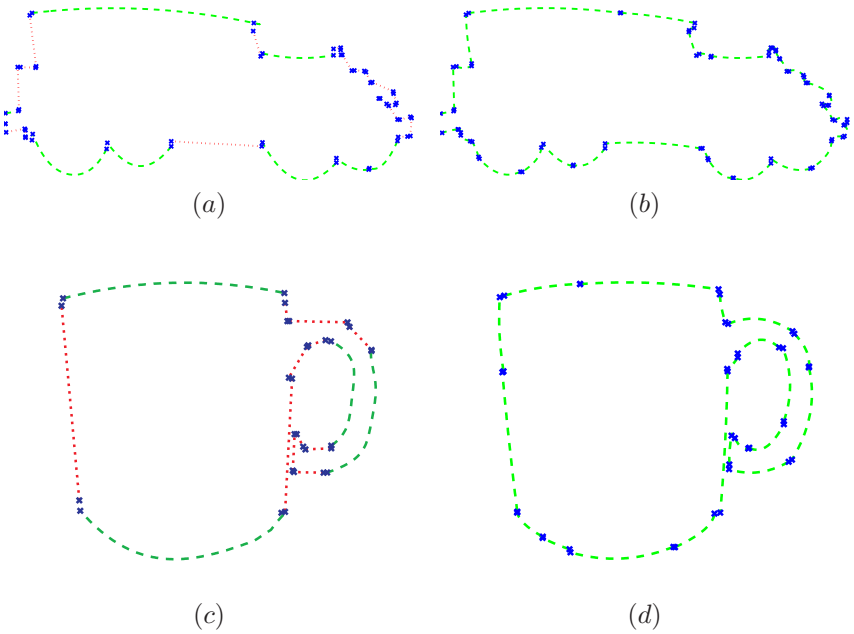


Figure 2.16: Figure (a) and (b) show the segmentation and approximation results on a car image using ACPF and NACPF, respectively. The red dotted lines and the green dashed lines correspond to first and second degree polynomials, respectively. Considering the result of ACPF, the contours are approximated accurately by polynomials of variable degree. Also, the segments are longer, which are more accurate approximations to the contour map, when compared to NACPF. Figure (c) and (d) show similar results on a cup image using ACPF and NACPF, respectively.

	Car		Cup	
	ACPF	NACPF	ACPF	NACPF
Average fitting cost per contour pixel	1.25	1.25	1.16	1.16
Average degree per contour pixel	1.84	2	1.76	2
Average length of the segments	53.54	39.42	90.44	52.23
Average Hausdorff distance per segment	2.36	2.92	3.58	3.83

Table 2.1: Values for the average fitting cost per contour pixel, the average degree per contour pixel, the average length of the segments and the average Hausdorff distance per segment, when comparing ACPF to NACPF, for images of the car and the cup in Figure 2.16.

We look at a few segmentation and approximation results on images of the MPEG-7 database. Figure 2.16 (a) and (b) show the results on a car using ACPF and NACPF, respectively. The red dotted lines and the green dashed lines correspond to first and second degree polynomial curves, respectively. Considering the result of ACPF, the contours are approximated accurately by polynomials of variable degree. It is satisfying to see that the wheels are separately approximated by parabolas, while the contours at the back of the car are approximated by straight lines. The endpoints of the segments correspond to the corners in the contour and are suitable as feature points. Values for the average fitting cost per contour pixel, the average degree per contour pixel, the average length of the segments and the average Hausdorff distance per segment can be found in table 2.1. From these values, we can conclude that ACPF gives longer segments, which are more accurate approximations to the contour map, when compared to NACPF. Similar results on a cup image using ACPF and NACPF are shown in Figure 2.16 (c) and (d), respectively.

2.6 Image segmentation into polynomial surfaces

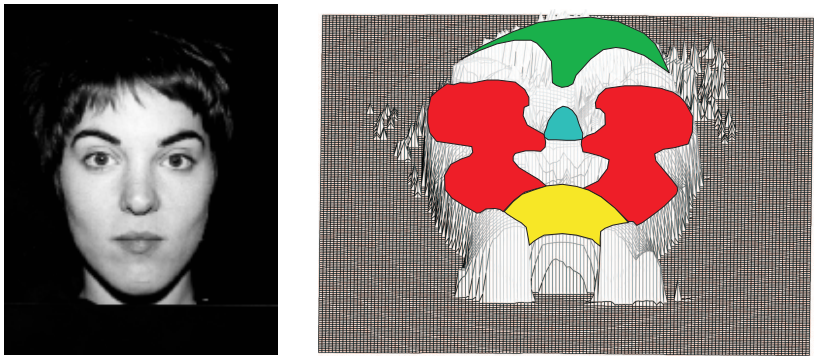
In this section, we describe how we segment a greyscale face image into smooth surface segments. This work was published in [Deboeverie et al., 2013b].

Adaptive region growing for image segmentation into polynomial surfaces is based on the same principles of local and global sampling of the region as in the previous section of contour segmentation into polynomial curves. However, mainly due to the different spatial ordering of contour pixels and image pixels, we present a modified version of adaptive region growing. Where region growing groups pixels along one dimension for connected contour pixels, region growing groups pixels along two dimensions for connected image pixels. Therefore, we present a different strategy when selecting elemental subsets. Moreover, because of different polynomial fitting functions, we present the modified computations for constructive fitting.

The segmentation is based on the property that, because of Lambert's cosine law [Lambert, 1760], when the light comes mainly from one direction, the intensity surface (image intensities) of a face image has the same shape as the skin surface itself. According to Lambert's law the light intensity observed from a diffusely reflecting surface is only determined by the angle between the surface normal and the direction of the incident light. Since skin is a relatively matte surface with uniform texture, we do not have to consider distinct types of material, illumination and shape as in [Maxwell and Shafer, 2000]. The head, which resembles a convex sphere with small concavities, will be seen as a collection of intensity patches of concave functions, and smaller patches of convex functions [Wagemans et al., 2010]. A function g is convex if $g(\lambda_1 p_1 + \lambda_2 p_2) \leq \lambda_1 g(p_1) + \lambda_2 g(p_2)$ for all convex combination $\lambda_1 + \lambda_2 = 1$, $\lambda_i \geq 0$. A function g is concave, if $-g$ is convex. Figure 2.17 (a) shows an example of a greyscale face image of the Stirling face database [Stirling,]. In this image, as is often the case, most of the light comes from a direction close to the viewing direction. Figure 2.17 (b) shows the grey values in 3-D, partially clustered in face segments, such as the forehead, the cheeks, the chin and the nose. These are convex body parts, seen as intensity patches of concave functions.

2.6.1 Constructive polynomial surface fitting

In this section, we present the mathematical foundations of constructive polynomial surface fitting on which the region growing processes in this work are based. The problem we study is that of finding a region of maximal size in which a surface can be approximated well by a polynomial function, given an initial seed region. We also consider the related problem of finding this region with minimal computational effort.



(a) (b)

Figure 2.17: (a): A greyscale image of the Stirling face database [Stirling,]. (b): The grey values in 3-D, partially clustered in face segments, such as the forehead, the cheeks, the chin and the nose. These are convex body parts, seen as intensity patches of concave functions.

In this work, we find regions of maximal size in which a low-degree polynomial approximation using a specified L_∞ fitting cost criterion exists. Let $f(x_i, y_i)$ represent the image intensities. Let G be a vector space of fitting functions, for instance, the vector space of bivariate polynomial functions of total degree d:

$$g(x,y) = \sum_{k=0}^{d} \sum_{l=0}^{k} \alpha_{l,k-l} x^l y^{k-l}, \tag{2.14}$$

where each polynomial is characterized by $n = (d+1)(d+2)/2$ coefficients $\alpha_{l,k-l}$.

The accuracy of fitting $g(x,y)$ over the segment S is measured with the L_∞ fitting cost. This fitting cost is defined as

$$r(S;g) = \max_{(x,y)\in S} |g(x,y) - f(x,y)|. \tag{2.15}$$

The best fit is the polynomial function $g(x,y)$ in G for which $r(S;g)$ is minimal. We denote this minimal cost as $r(S)$, i.e,

$$r(S) = \min_{g\in G} r(S;g). \tag{2.16}$$

The L_∞ fitting cost over any segment S can be estimated (but not computed exactly) very efficiently in terms of so-called elemental subsets [Veelaert, 1997, Veelaert and Teelen, 2006]. Elemental subsets of cardinality m are subsets of S

that contain precisely $n+1$ pixels. Introducing elemental subsets will bring the advantage of minimizing the time to compute the fittings costs when adding new pixels to large segments during a region growing process. The importance of an elemental subset lies in the fact that the fitting cost over an elemental subset can be computed in a straightforward manner. Let $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be an elemental subset. Let E_j denote the cofactor (signed minor) of the element at the intersection of the last column and the jth row of the following matrix:

$$(A_D|B_D) = \left(\begin{array}{ccccccc|c} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & \cdots & f(x_1,y_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & y_m & x_m^2 & x_my_m & y_m^2 & \cdots & f(x_m,y_m) \end{array}\right) \tag{2.17}$$

Then one can show that the fitting cost over an elemental subset D can be computed by

$$\begin{aligned} r(D) &= \frac{\det(A_D|B_D)}{|E_1| + \ldots + |E_m|} \\ &= \frac{|E_1 f(x_1,y_1) + \ldots + E_m f(x_m,y_m)|}{|E_1| + \ldots + |E_m|}, \end{aligned} \tag{2.18}$$

provided the denominator is non-vanishing [Veelaert, 1997, Veelaert and Teelen, 2006]. Furthermore, the fitting cost over any segment S that contains more than m pixels is [Veelaert, 1997, Veelaert and Teelen, 2006]:

$$r(S) = \max_{D \in U} r(D) \tag{2.19}$$

where U is the collection of all elemental subsets D of S for which $|E_1| + \ldots + |E_m| > 0$. Expression (2.19) holds when U is non-empty, which is the case as soon as not all the pixels of the segment lie exactly on a common curve, $g(x, y) = 0$.

Figure 2.18 shows an example of the pixels of an elemental subset (grey squares) of a segment S (grey and white squares) of the forehead. In this example, the fitting costs are computed for polynomial functions of degree $d = 2$. Consequently, the number of pixels in the elemental subset is $m = 7$.

In principle, to find $r(S)$ we must evaluate $r(D)$ over all possible elemental subsets of S, a collection that grows as $O(|S|^m)$ when the segment grows larger. However, we can obtain a reliable estimate of the fitting cost with far fewer computations than those required for computing the fitting cost exactly. The fitting cost of a data set can be estimated very reliably from a few of its elemental subsets [Veelaert, 1997, Veelaert and Teelen, 2006]. Instead of calculating $r(S)$, we compute the estimate

$$\tilde{r}(S) = \max_{D \in \tilde{W}} r(D), \tag{2.20}$$

where $\tilde{W}$ forms a rigid subcollection of M elemental subsets of S [Veelaert, 1997, Veelaert and Teelen, 2006]. In the experiments, we achieve reliable estimation

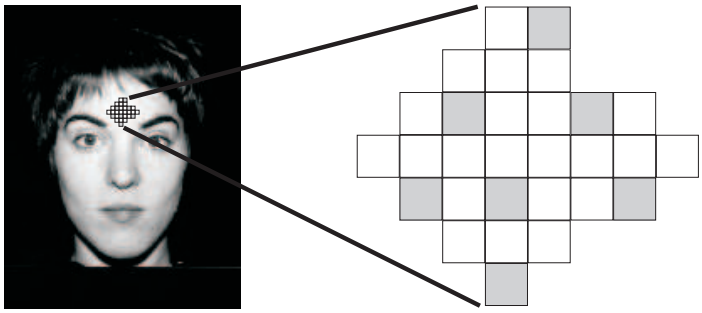


Figure 2.18: An example of the pixels of an elemental subset (grey squares) of a segment S (grey and white squares) of the forehead.

of the fitting cost by randomly selecting a small fixed number (e.g. $M = 10$) of elemental subsets D. Decreasing the number of elemental subsets will decrease the computation time. However, the fitting cost will be estimated less accurately. Since the number of elemental subsets used to estimate fitting costs is constant during the entire region growing process, the resulting algorithms have linear time complexity. This means that the time to add a pixel to a segment is constant, although the number of pixels to which a new pixel has to be compared grows.

During region growing, the L_∞ fitting cost shows direct response to intensity discontinuities, since the L_∞ norm looks at the maximum deviation between pixel values and the fitting polynomial. This is advantageous in region growing when making decisions about adding a new pixel, discarding an outlier or stopping at an edge. For each pixel to be added in region growing, the fitting cost in Eq. (2.20) is computed when fitting a low-degree polynomial surface to the pixel and the segment. The fitting cost is an indicator of whether the pixel belongs to the segment according to the polynomial surface. It is computed without computing the actual best fitting polynomial. The best fit has only to be computed when the segment is finished.

The region growing algorithm, which is described in the following section, uses adaptive thresholding of the L_∞ fitting cost $\tilde{r}(S)$ as a segmentation criterion.

2.6.2 Surface segmentation with adaptive region growing

In this section, we describe the algorithm to segment a greyscale image into smooth surface segments with adaptive region growing based on low-degree polynomial fitting. We explain how the region growing is made adaptive to the local image properties, such that surface segments are represented as polynomial surfaces with a variable polynomial degree and a variable fitting error.

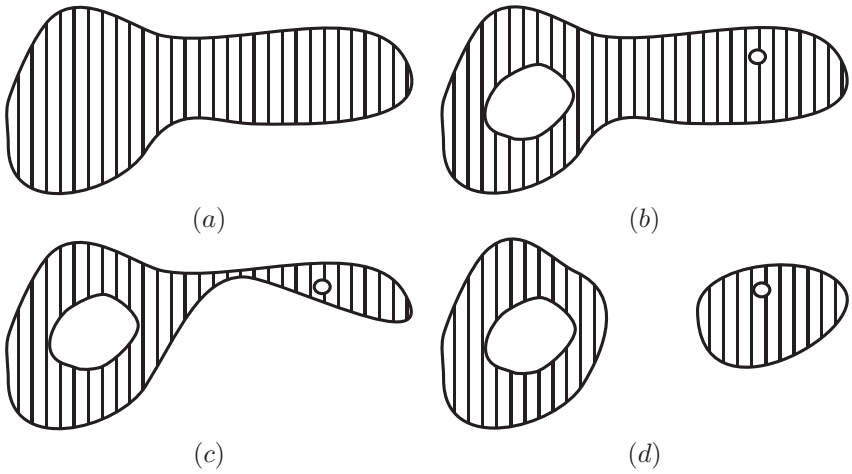


Figure 2.19: Topologies of region growing in our approach. (a): Region growing groups connected pixels. (b): Region growing can grow around outliers and larger regions. (c) Region growing can group pixels along a thin region. (d): Region growing cannot group disconnected pixels.

Computing the fitting cost with elemental subsets boils down to a sampling of the image region. Sampling allows us to find out how each pixel in an image region contributes to the fitting cost when fitting a polynomial to that region. Sampling with elemental subsets and region growing fit well together, since they both treat regions with connected pixels and a closed contour. Figure 2.19 (a) shows a topology of region with connected pixels and a closed contour. Region growing in this work can also treat the topologies in Figures 2.19 (b) and (c), where pixels are still connected and contours are still closed, but where region growing has grown around outliers and larger regions, or where region growing has grown along a thin region, respectively. Figures 2.19 (d) shows a topology which cannot be treated by region growing because pixels are disconnected.

Also in 2D case, fitting costs can be computed for local regions as well as for a global region, where the corresponding global fitting cost is calculated from the corresponding local fitting costs. The terms local and global refer to a part of the segment and the entire segment, respectively. Moreover, the local and global fitting costs can be combined in several ways. In the example in Figure 2.20, we use the same local fitting cost when fitting a polynomial to the pixels of the regions S_1 and S_2. This local cost is computed once and is used in estimating the cost of the region S_3. In fact, for the estimated fitting cost we have $\tilde{r}(S_3) = \max\left\{\max_{D\in\tilde{W}_2} r(D), \max_{D\in\tilde{W}_1} r(D)\right\}$, where $\tilde{W}_i$ corresponds to S_i and $S_1 \subset S_2 \subset$

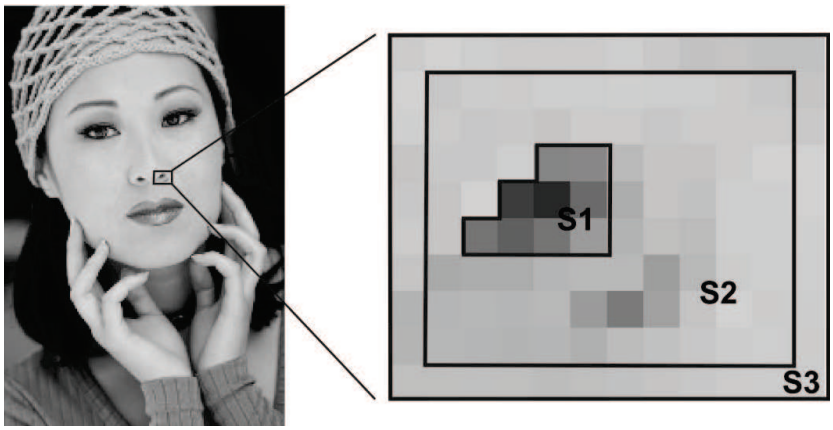


Figure 2.20: Computing the fitting cost with elemental subsets allows to find out how each pixel in an image region contributes to the fitting cost when fitting a polynomial. In this example, we compute the local fitting cost when fitting a polynomial to the pixels of the regions S_1 and S_2. At the same time, this local fitting cost is part of the global fitting cost when fitting a polynomial to the pixels of the region S_3.

S_3.

In this work, we demonstrate the key idea of combining local and global fitting costs with a strategy for image segmentation in a region growing process with adaptive thresholding. The proposed region growing method examines the thresholding of local fitting costs to decide if a new pixel is to be added to a segment (to identify edges and outliers), while the thresholding of global fitting costs controls if the polynomial degree is adapted. This will become clear when we explain the region growing based on the segmentation of the image surface in Figure 2.21. Segmenting this image surface covers all the possible region growing problems we have to deal with. Five items are considered.

Item 1: Start region growing

Region growing starts with a seed pixel, and then repeatedly adds new pixels to the segment, as long as the segmentation criterion is still satisfied on the enlarged segment. Seed pixels are chosen as local grey value extrema of the image and where the gradient remains small (e.g. grey value extrema in a 30 by 30 neighbourhood and a gradient magnitude < 0.5). This avoids the selection of seed pixels at an edge, since seed pixels at an edge offer fewer opportunities to grow. For each new seed pixel we start with the polynomial degree set equal to zero in Eq. (2.14). Item 1 in Figure 2.21 shows an example of such a seed pixel. Pixels are then added

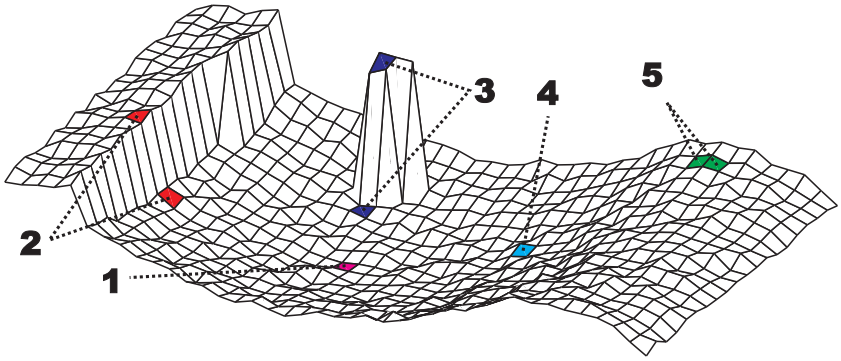


Figure 2.21: An example of image intensities which we segment into smooth surface segments with adaptive region growing and polynomial fitting. Item 1: Seed pixels are chosen as local extrema of the image. Item 2: The region growing stops at edges. Item 3: The algorithm grows around outliers. Edge pixels and outliers are not added to the segment because that would cause too fast and too direct increases in the L_∞ fitting cost. Item 4: The polynomial degree is adapted when the variation of grey values is smooth, but too large to capture it with the current polynomial degree. Item 5: The segmentation starts with a new segment when the variation of grey values becomes too large to capture it with a flat, planar, convex, concave or saddle like surface.

one by one. Pixels to be added are selected from pixels next to the boundaries of the segment using morphological dilation with a circular structuring element.

Item 2 and 3: Stop at edge pixels and grow around outliers

For each pixel p_i already added to the segment $S_{i-1} = \{p_0, p_1, \ldots, p_{i-1}\}$, the region growing keeps track of the fitting cost when fitting a low-degree polynomial surface to S_{i-1} and p_i. To this end we define

$$\tilde{r}(S_{i-1}; p_i)$$

as the cost of adding p_i to S_{i-1}.

The decision to add a new pixel p_k is then based on

$$X_k = \tilde{r}(S_{k-1}; p_k) - \frac{1}{|R_k|} \sum_{p_i \in R_k} \tilde{r}(S_{i-1}; p_i) \leq T_X, \tag{2.21}$$

which measures the local behaviour of the consecutive fitting costs. The term local refers to a small part of the segment S_k. A segment will grow until the local variation of the grey values change, giving rise to discontinuities in the consecutive fitting costs. However, to avoid overreaction to discontinuities due to noise and

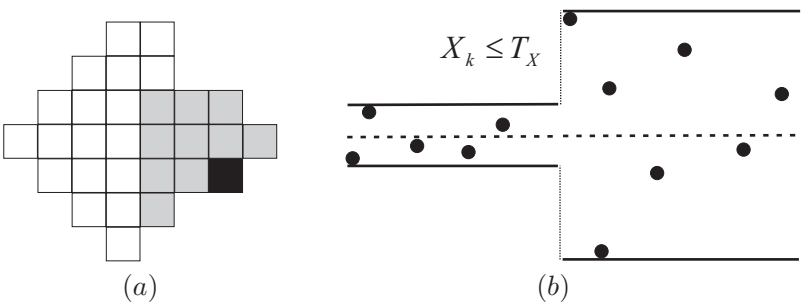


Figure 2.22: (a): An example of a new pixel p_k (black square) and a local neighbourhood R_k (grey squares) in the segment S_{k-1} (white and grey squares). (b): 1-D example of a strong transition in the variation of pixels, which causes a fast and direct increase in the L_∞ fitting costs. A pixel is added to the segment if the local behaviour X_k of the fitting costs is lower than the threshold T_X.

small speckles, the increase of the fitting cost is compared to the mean of the previous fitting costs in a local neighbourhood R_k of p_k (e.g. a 2 pixel deep square neighbourhood). Figure 2.22 (a) shows an example of p_k (black square) and R_k (grey squares) in S_{k-1} (white and grey squares). The new pixel p_k is added to S_{k-1} when X_k is lower than the threshold T_X. When X_k exceeds T_X, i.e. when adding p_k would increase the fitting cost significantly more than on average, p_k is not added to S_{k-1}. Figure 2.22 (b) shows a 1-D example of a strong transition in the variation of pixels, which causes a fast and direct increase in the L_∞ fitting costs. This occurs when p_k is an outlier or lies on an edge. If p_k is an outlier the segment will grow around p_k. Items 2 and 3 in Figure 2.21 show examples of such an edge and outlier, respectively.

Thus, as a stopping criterion during region growing we examine the local differences between the fitting costs, not their magnitude. Discontinuities (outliers and edges) are distinguished from gently rising variation.

Item 4: Adapt the polynomial degree

The decision to increase the degree of the polynomial surfaces is based on

$$Y_k = \frac{1}{|B_k|} \sum_{p_i \in B_k} \tilde{r}(S_{i-1}; p_i) \leq T_Y, \tag{2.22}$$

which measures the global behaviour of the fitting costs. The term global refers to the entire segment S_k. It is determined by the mean of the fitting costs when fitting a low-degree polynomial surface to the segment S_{i-1} and the pixels p_i on the boundary B_k. These fitting costs contain all recent information (fitting costs which were computed last) about the maximum deviation between the pixels of

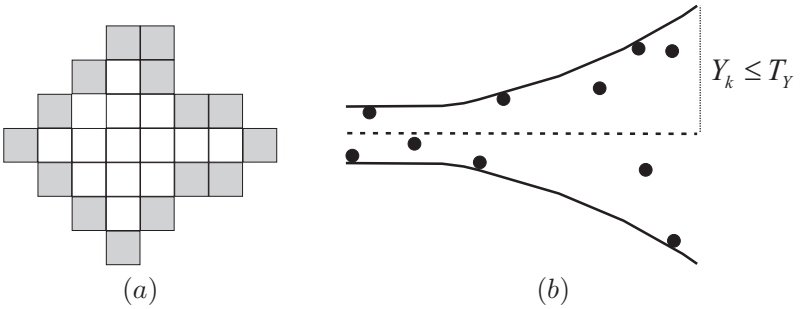


Figure 2.23: (a): An example of the pixels p_i on the boundary B_k (grey squares) in the segment S_k (white and grey squares). (b): 1-D example of a smooth transition of pixels, which causes a smooth increase in the L_∞ fitting costs. The polynomial degree is increased by one, when the global behaviour Y_k of the fitting costs exceeds the threshold T_Y.

S_k and the polynomial surface. B_k grows when S_k gets larger. Only the fitting costs which were computed when adding pixels of B_k are considered to increase the polynomial degree, since for previous added pixels Eq. (2.22) was already met. Furthermore, we consider the fitting costs which were computed when adding all pixels of B_k, since we do not allow to let the addition of only one pixel influence the polynomial degree. Figure 2.23 (a) shows an example of the pixels p_i on B_k (grey squares) in S_k (white and grey squares). The polynomial degree is increased by one, when the global variation of grey values is becoming too large. This is when Y_k exceeds the threshold T_Y. Then, Y_k is recomputed for this new degree. In fact, the degree increases until either Y_k is again within the limit of the threshold T_Y, or a maximum degree is exceeded. A maximal polynomial degree of two is sufficient to expand the segment along smooth flat, planar, convex, concave and saddle intensity functions. Figure 2.23 (b) shows a 1-D example of a smooth transition of pixels, which causes a smooth increase in the L_∞ fitting costs. Item 4 in Figure 2.21 shows a pixel where the variation of grey values is smooth, but too large to capture it with the current polynomial degree.

Thus, for the adaption of the polynomial degree we examine the global behaviour of the fitting costs. The global behaviour at the boundary reveals whether a smooth segment is slowly evolving towards either a flat, planar, convex, concave or saddle surface.

Algorithm 1 describes the apdaptive region growing process, as discussed above.

Algorithm 1 Adaptive Region Growing

input: a greyscale image
output: a list of segments
begin
  while new seed pixel p_s **do**
    new segment $S_s \leftarrow \{p_s\}$
    polynomial degree $d \leftarrow 0$
    while new pixel p_k to be added **do**

      compute fitting cost $\tilde{r}(S_{k-1}; p_k) \leftarrow \max_{D \in \tilde{W}} r(D)$

      update local measure:
      $X_k \leftarrow \tilde{r}(S_{k-1}; p_k) - \frac{1}{|R_k|} \sum_{p_i \in R_k} \tilde{r}(S_{i-1}; p_i)$

      if $X_k \leq T_X$ **then**
        add pixel to segment $S_k \leftarrow S_{k-1} \cup \{p_k\}$
      end if

      update global measure:
      $Y_k \leftarrow \frac{1}{|B_k|} \sum_{p_i \in B_k} \tilde{r}(S_{i-1}; p_i)$

      while criterion to adapt the degree, i.e. $Y_k > T_Y$ **do**
        increase degree $d \leftarrow d + 1$
        recompute Y_k for new degree
      end while
    end while
  end while
end

Item 5: Finish segment

When no more pixels can be added along the boundary, the segment is completed, and the segmentation process starts a new segment at a new seed pixel. Item 5 in Figure 2.21 shows a pixel where a new segment is created, because the variation of grey values becomes too large to capture it with a single flat, planar, convex, concave or saddle like surface.

The growing process has been designed to find smooth segments in an image. It will grow around outliers and it will stop at edges. However, there is no guarantee that two segments sharing a common boundary will stop exactly at the same edge. The treatment of this issue requires some additional processing.

To force segments to stop at the same common boundary we do not allow the segments to grow over pixels where the Gradient is strong, i.e. strong edges of an edge map. The Canny edge detector [Canny, 1986] is suitable, because it results in

(a) (b)

Figure 2.24: Image segmentation result by region growing without post-processing. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. (a) A segmented face image by region growing without stopping at canny edges. The region growing finds sometimes segments that do not stop at the real face contours. In these cases, we find additional segments to the real face contours. (b): A segmented face image without morphological closing and segment merging. The region growing finds a lot of small segments.

thin edges of one pixel thickness, similar to the segment boundaries in the region growing. In addition to the edges of the edge detector, the segmentation will find additional edges at the segment boundaries, which are much smoother than the edges found by an edge detector. These additional edges are necessary in the segmentation. They represent smooth meaningful transitions in a surface, e.g., the gradual transition from a concave surface into a convex surface. Figure 2.24 (a) shows a segmented face image by region growing without stopping at the edges of the Canny edge map. The region growing finds sometimes segments that do not stop at the real face contours. In these cases, we find additional segments to the real face contours.

Since large segments in an image often represent important object features, at a smooth transition a new segment A is allowed to grab pixels from an existing segment B. For this to happen, we require that A must already be larger than B and the polynomial degree of A must not be higher than the polynomial degree of B, since for a higher polynomial degree the fitting cost has a higher probability to

Figure 2.25: This figure shows a segmented face image with image size 142x182. The face is segmented in 23 surface segments. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively.

be lower.

Finally, morphological closing is used to fill the gaps in the segments. Thus outliers, for which $X_k > T_X$, but which are enclosed by a segment, are added to the segment. Similarly, we prevent the segmentation to result in many small segments (e.g. ≤ 10 pixels). Therefore, if possible, we add small segments to larger adjacent segments under less strong conditions (e.g. $X_k \leq 1.2T_X$). Figure 2.24 (b) shows a segmented face image without morphological closing and segment merging. The region growing finds a lot of small segments.

Figure 2.25 shows a segmented face image with image size 142x182. The face is segmented in 23 surface segments. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. Many surface segments correspond to meaningful parts of the face, such as the cheeks, the forehead, the chin and the eyebrows. Examples of additional smooth edges are the edges separating the cheeks and the chin.

2.6.3 Best fit polynomial surface - Surface Intensity Map

Our face segmentation with adaptive region growing results in a face which is divided into several surface segments. In this segmentation each surface segment can be approximated by a low-degree polynomial surface as in Eq. (2.14). Then,

the image intensities of a segment can be computed from the polynomial coefficients. The coefficients are computed after all segments have been found and post-processed. Remember that during region growing, only the fitting cost was computed. The best fit can be computed with an L_2 fit or an L_∞ fit. A method for computing the best L_∞ fit is described here.

The best L_∞ fit is defined as

$$g_{\min} = \text{argmin}_{g \in G} r(S; g).$$

In [Veelaert and Teelen, 2006] it has been shown that

$$g_{\min} = \text{argmin}_{g \in G} r(D_{\max}; g)$$

where $D_{\max}$ is the elemental subset in S with the largest cost, i.e.,

$$D_{\max} = \text{argmax}_{D \subseteq S} r(D).$$

Thus the computation of the best fit consists of two steps. First, we have to determine the elemental subset D_0 of maximum cost.

Computing the fitting cost for all elemental subsets in S would be very computationally expensive. Therefore we propose an alternative algorithm described in [Veelaert, 2012], which starts with an arbitrary subset D_j with fitting cost $r(D_j)$. For this subset we compute the best fit $g^j_{\min}(x, y)$. Let $\epsilon_j = r(D_j)$, then the functions $g^j_{\min}(x, y) - \epsilon_j$ and $g^j_{\min}(x, y) + \epsilon_j$ are called the two supporting polynomials of the elemental subset. It can be shown that the image intensity at each pixel (x_i, y_i) in D_j either satisfies $f(x_i, y_i) = g^j_{\min}(x_i, y_i) - \epsilon_j$ or $f(x_i, y_i) = g^j_{\min}(x_i, y_i) + \epsilon_j$. If D_j is not the elemental subset with largest cost, there is at least one pixel (x_k, y_k) in S for which we either have $f(x_k, y_k) < g^j_{\min}(x_k, y_k) - \epsilon_j$ or $f(x_k, y_k) > g^j_{\min}(x_k, y_k) + \epsilon_j$. It follows that $D_j \cup \{(x_k, y_k)\}$ contains at least one elemental subset D_{j+1} with larger cost than D_j. Here are $n + 2$ of these subsets D'_k of $D_j \cup \{(x_k, y_k)\}$. Thus the elemental subset with largest cost can be found iteratively, by selecting an arbitrary elemental subset, computing its supporting surfaces. If there is still a pixel not enclosed by the supporting surfaces, we can immediately find a new elemental subset D'_k with larger cost.

Furthermore, one can show that the best fit $g^j_{\min}$ can be easily found by solving the system of linear equations

$$g^j_{\min}(x_i, y_i) = f(x_i, y_i) + \text{sign}(E_i)\epsilon_j, \quad \forall (x_i, y_i) \in D_j$$

where $\text{sign}(E_i)$ is the sign of the cofactor of the fitting matrix in Eq. (2.17). In fact, these signs determine whether a pixel $(x_i, y_i, f(x_i, y_i))$ either lies on the supporting surface $g^j_{\min} + \epsilon_j = 0$ or the supporting surface $g^j_{\min} - \epsilon_j = 0$.

We introduce a compact feature, the Surface Intensity Map (SIM), which integrates structural information and spatial information by grouping pixels of an

Figure 2.26: The best fit polynomial surfaces for the surface segments in Figure 2.25

intensity map into surface segments. It describes polynomial surfaces with a flat, planar, convex, concave or saddle shape in a surface map. Figure 2.26 shows the best fit polynomial surfaces for the surface segments in Figure 2.25. The image size is 142x182. The face is represented by 23 polynomial surfaces. The result is an approximated image of the original image in Figure 2.17 (a). It is satisfying to see that the facial features are nicely reconstructed from the low-degree polynomial surfaces.

2.6.4 Curvature of polynomial surfaces

Our face model uses the curvatures (flat, planar, convex, concave or saddle like behaviour) of the polynomial surfaces to perform face analysis.

Convex, concave or saddle like behaviour of a second-degree polynomial surface $g(x, y)$ as in Eq. (2.14) is defined by the signs of the eigenvalues of the Hessian matrix:

$$H(g) = \begin{bmatrix} \frac{\partial^2 g}{\partial^2 x} & \frac{\partial^2 g}{\partial x \partial y} \\ \frac{\partial^2 g}{\partial x \partial y} & \frac{\partial^2 g}{\partial^2 y} \end{bmatrix}. \tag{2.23}$$

The entries of the matrix $H(g)$ are the second order derivatives of the surface with respect to x and y coordinates. For a quadratic surface, the second derivatives are constant and hence $H(g)$ is independent of the location of the pixel in the segment.

From Eq. (2.14), we find

$$H(g) = \begin{bmatrix} 2\alpha_{2,0} & \alpha_{1,1} \\ \alpha_{1,1} & 2\alpha_{0,2} \end{bmatrix}. \tag{2.24}$$

The maximum and minimum curvatures are determined by the eigenvalues of this matrix, which are found by solving the following characteristic equation:

$$Hk = \lambda k. \tag{2.25}$$

The homogeneous system $(H - \lambda I)k = 0$ has a non-zero solution if the determinant of its coefficient matrix is zero:

$$\begin{vmatrix} 2\alpha_{2,0} - \lambda & \alpha_{1,1} \\ \alpha_{1,1} & 2\alpha_{0,2} - \lambda \end{vmatrix} = 0. \tag{2.26}$$

The matrix H is symmetric, hence the solution yields two real values λ_1 and λ_2. Both λ_1 and λ_2 are positive for a convex surface and negative for a concave surface. Eigenvalues have opposite signs for a saddle surface. One of the eigenvalues is zero for a cylindrical surface.

The product of the two eigenvalues gives the Gaussian curvature $G = \lambda_1\lambda_2$. The Gaussian curvature is an intrinsic measure of curvature, i.e., its value depends only on how distances are measured on the surface, not on the way it is isometrically embedded in space. The Gaussian curvature is zero when one of the eigenvalues is zero, which corresponds to a cylindrical surface, as when both are zero, which indicates a plane.

The eigenvectors from the characteristic Eq. (2.26) point to the direction of maximum and minimum curvatures. The azimuth of maximum curvature θ is given by

$$\theta = \arctan\left(\frac{k_{11}}{k_{12}}\right), \tag{2.27}$$

where k_{11} and k_{12} are components of the eigenvector corresponding to the largest eigenvalue. The direction of minimum curvature is orthogonal to the direction of maximum curvature.

An example of the curvatures of polynomial surfaces in a face images is shown in Figure 2.27. The magenta, cyan and yellow colours correspond to convex, concave and saddle like behaviour, respectively. We find concave polynomial surfaces for the forehead, the cheeks, the chin and the nose, while the throat is a saddle surface. Convex, concave and saddle like behaviour can be disturbed by specular reflection on the skin, or by diffuse light coming from all directions, we have found that in normal circumstances the effect is sufficiently strong to find segments that correspond to meaningful parts of the face.

Figure 2.27: The convex, concave or saddle like behaviour of the polynomial surfaces, indicated by the colours magenta, cyan and yellow, respectively.

2.6.5 Evaluation of image segmentation and approximation

In this section we evaluate our segmentation technique on the AR face database [Martinez and Benavente, 1998] and the Berkeley segmentation dataset and benchmark (BSDS300) [Martin et al., 2001].

Most of the computation time of the region growing method is spent on the calculation of the fitting cost $\tilde{r}$. We developed a partially optimized program, which is implemented in C++ and running on a 2.8 GHz processor, 4 GB RAM and 64-bit operating system. For images of size 200x200, the computation time of our program is approximately 1.0 second per image. As a comparison, the computation time of the power watershed algorithm [Couprie et al., 2011] is approximately 1.2 seconds per image.

To find the optimal parameter set of adaptive region growing, we measure the image approximation accuracy with a surface area weighted mean of the L_∞ fitting costs $\tilde{r}(S)$ of the polynomial surfaces . Figure 2.28 shows segmented and approximated face images with different approximation accuracies. The mean fitting cost varies from 0.83 at the upper left, to 9.53 at the lower right. A high approximation accuracy (low mean fitting cost) gives a high number of smaller segments, providing a good approximation quality. On the other hand, a low approximation accuracy (high mean fitting cost) gives a low number of larger segments, providing approximation quality less well. Depending on the desired purpose (approximation or segmentation), one has to find a good balance between the size of the

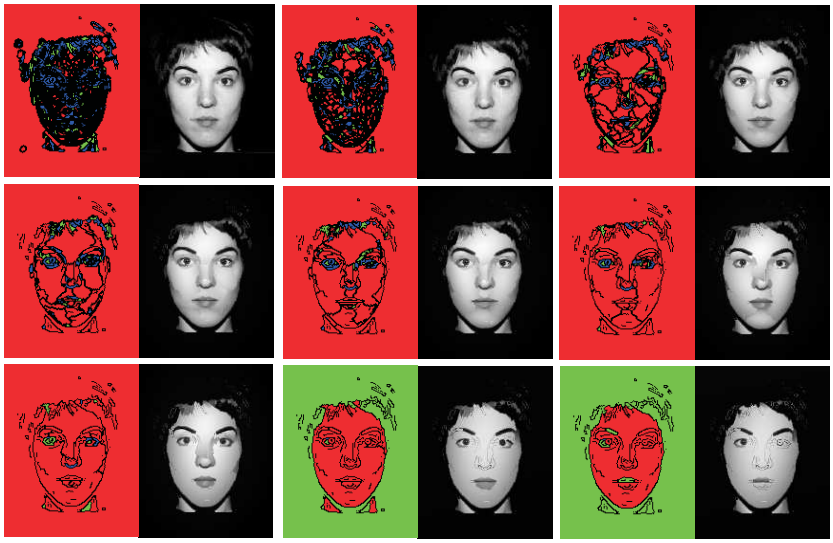


Figure 2.28: Segmented and approximated images with different approximation accuracies. The mean fitting cost varies from 0.83 at the upper left, to 9.53 at the lower right.

segments and the quality of the approximated images.

To demonstrate the advantages of using the L_∞ norm in a region growing process, we apply a variation of our method, where the L_2 norm is used instead of the L_∞ norm. Figures 2.29 (a) and (b) show segmented face approximations produced by L_2 and L_∞ based adaptive region growing, respectively. In comparison with the L_∞ norm, we can clearly ascertain that the segment boundaries are more arbitrary and that the segment approximations are less accurate when using the L_2 norm. This confirms that the usage of the L_∞ fitting cost in a region growing process allows us to make more accurate decisions about adding a new pixel, discarding an outlier or stopping at an edge. Note that, despite the limitations of the L_2 norm, the proposed strategy for image segmentation with adaptive thresholding is still successful.

Face image segmentation

We visually compare several segmentation techniques on a face image. Columns (b), (c), (d) and (e) on the second row in Figure 2.30 show segmented face approximations of the face image in Figure 2.30 (a), produced by normalized cuts [Shi and Malik, 2000], mean shift [Comaniciu and Meer, 2002], power watersheds [Couprie et al., 2011] and the proposed segmentation technique based on a polynomial surface model, respectively. The power watershed method is a seeded

(a) (b)

Figure 2.29: (a) and (b): segmented face approximations produced by L_2 and L_∞ based adaptive region growing, respectively. In comparison with the L_∞ norm, we can clearly ascertain that the segment boundaries are more arbitrary and that the segment approximations are less accurate when using the L_2 norm.

image segmentation algorithm that includes the graph cuts, random walker, and shortest path optimization algorithms. In this comparison, the power watershed algorithm uses the same seed pixels as produced by the proposed method. We can clearly ascertain that in contrast to the proposed method, the segments produced by existing segmentation techniques do not always coincide with facial features and the contours separating the surface segments often do not correspond to real image face edges. Furthermore, the polynomial segment approximations do not accurately reconstruct the face image.

The AR face database consists of two series of thirteen face images of 136 persons under various circumstances, with 76 males and 60 females. The face images have an image size of 192x144. For this dataset, we set the segmentation parameters $T_X = 1.0$ and $T_Y = 2.0$, preserving a good balance between the size of the segments and the approximation quality. These parameters have been manually tuned on a small number of images. Segmentation results on the AR face database are shown in Figure 2.31. The columns (a), (b) and (c) show the original greyscale images, the segmented images and the approximated images, respectively. The blue, green and red colours in the segmented images correspond to zero, first and second degree polynomial surfaces, respectively. Table 2.2 gives an overview of the mean and standard deviation of the computation time per image,

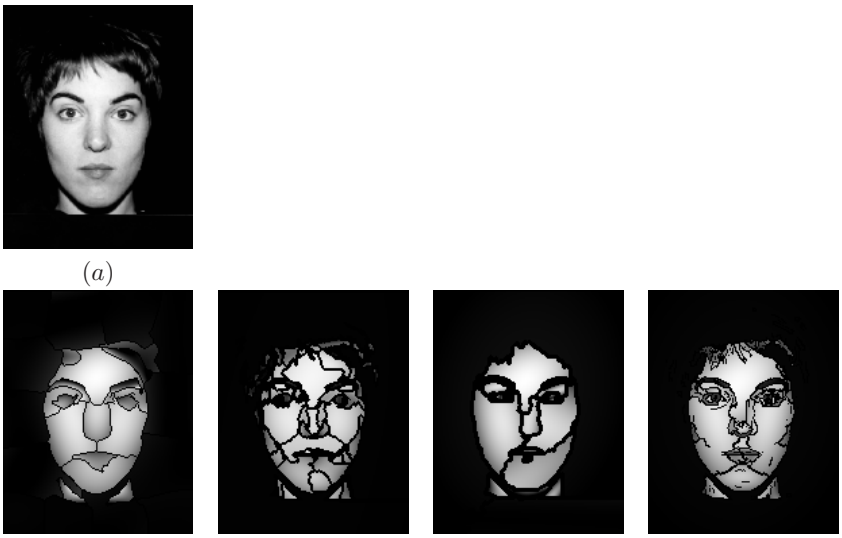


Figure 2.30: Visual comparison of image segmentation techniques on a face image. Columns (b), (c), (d) and (e) on the second row show segmented face approximations of the face image in (a), produced by normalized cuts [Shi and Malik, 2000], mean shift [Comaniciu and Meer, 2002], power watersheds [Couprie et al., 2011] and the proposed segmentation based on a polynomial surface model, respectively. In contrast to existing segmentation techniques, the segments produced by the proposed method correspond more to facial features, the contours separating the surface segments coincide with real image face edges and the low-degree polynomial approximations accurately reconstruct the face image.

the mean fitting cost (2.6.5) and the number of surfaces, respectively.

The graph in Figure 2.32 plots the numbers of surface segments when segmenting images of the AR face database in function of mean fitting costs (2.6.5). We conclude that for mean fitting costs of two or higher, which corresponds to low approximation accuracies, the mean numbers of surfaces and curves is relatively constant. This means that there is a small stable set of large segments. Note that for these approximation accuracies, the faces in the approximated images are still recognizable. For mean fitting costs of two or lower, which corresponds to high approximation accuracies, the mean number of surfaces and curves grows exponentially. This means that there are many small surface segments. Note that the introduction of an edge map, over which the surface segments cannot grow, is responsible for a lower limit on the number of segments for higher fitting costs.

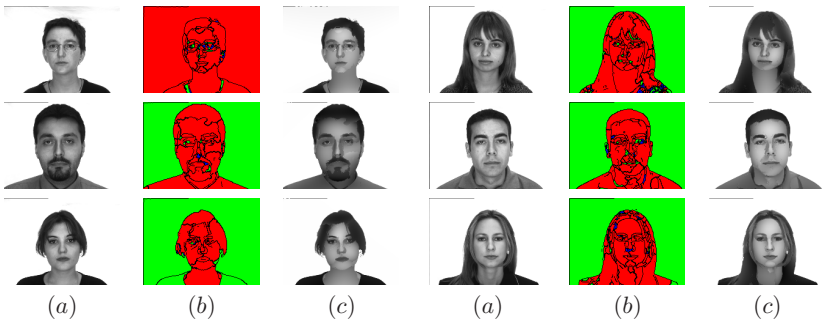


Figure 2.31: (a): The original greyscale images. (b): The segmented images. (c): The approximated images.

AR face database [Martinez and Benavente, 1998]	performance statistics
image size	192x144
computation time (ms)	968 ± 121
mean fitting cost	2.04 ± 0.63
#surfaces	11.20 ± 4.49

Table 2.2: This table gives an overview of the mean and standard deviation of the computation time, the mean fitting cost and the number of surfaces, respectively.

General image segmentation

The BSDS300 consists of 300 natural images, delivered with ground truth human annotations. To compare a segmentation with multiple ground truth images we examine the Probabilistic Rand Index (PRI) [Pantofaru and Hebert, 2005]. The PRI measures the fraction of pixel pairs, whose labels are consistent in the test segmentation and the ground truth one. The PRI averages over multiple ground truth segmentations and takes values in the interval [0,1], where 0 means that the acquired segmentation has no similarities with the ground truth and 1 means that the test and ground truth segmentations are identical.

Segmentation results on the BSDS300 are shown in Figure 2.33. For this dataset, we set the segmentation parameters $T_X = 1.3$ and $T_Y = 2.8$. As we will show further on in the results, these parameters result in a maximum PRI. The columns (a), (b), (c) and (d) show the original greyscale images, the segmented images, the approximated images and the images with an indication of the convex, concave or saddle like behaviour, respectively. The PRIs are indicated under the images. The blue, green and red colours in the images in column (b) correspond to zero, first and second degree polynomial surfaces, respectively. We ascertain that many surface segments correspond to meaningful parts of the image. The

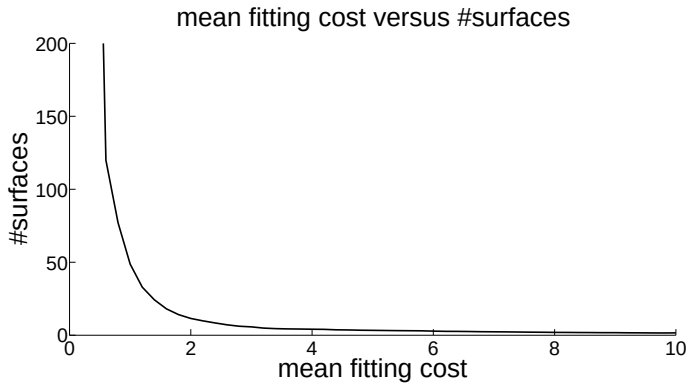


Figure 2.32: This graph plots the numbers of surface segments when segmenting images of the AR face database in function of mean fitting costs. For low mean fitting costs, there is a small stable set of large segments.

polynomial surfaces in the images in column (c) provide good approximation of the image, while preserving all the necessary details of the objects in the approximated images. In the images in column (d), the cyan, magenta and yellow colours correspond to concave, convex and saddle surfaces, respectively. For instance in the face image, the head is a convex body part, seen as an intensity patch of a concave function. This rough segment classification can serve as an input for a face detection algorithm.

Table 2.3 gives an overview of the mean and standard deviation of the computation time, the mean fitting cost, the number of surfaces and the PRI, respectively. When considering the number of surfaces, we find that our technique divides an image in a small number of surface segments. The graph in Figure 2.34 plots the numbers of surface segments when segmenting images of the BSDS300 in function of mean fitting costs. We find that for mean fitting costs of 3 or higher, which corresponds to low approximation accuracies, the mean numbers of surface segments decreases slowly. This means that there is a small stable set of large segments. In contrast, for mean fitting costs of 3 or lower, which corresponds to high approximation accuracies, the mean number of surface segments grows exponentially. This means that there are many small segments.

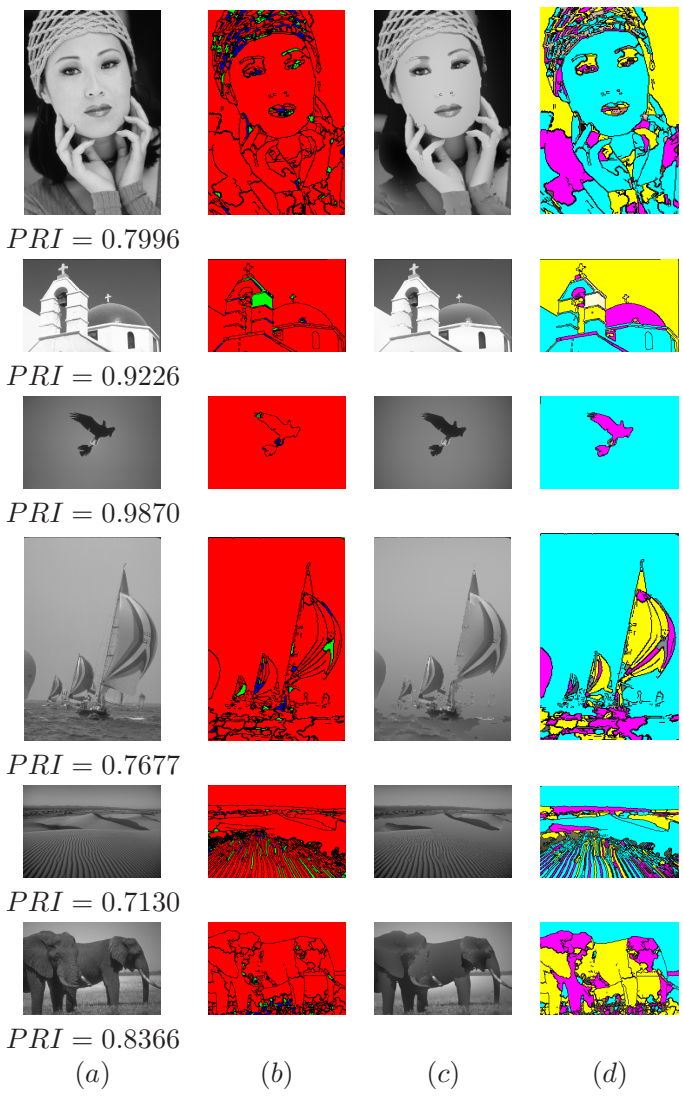


Figure 2.33: (a): Greyscale images with image size 240x160. (b): The segmented images with the corresponding PRIs. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. Many surface segments correspond to meaningful parts of the image. (c): The surface approximated images. Objects are nicely approximated by the low-degree polynomial surfaces. (d): The convex, concave or saddle like behaviour of the second degree polynomial surfaces, indicated by the colours magenta, cyan and yellow, respectively. Convex object parts with diffuse reflecting surfaces are seen as intensity patches of a concave functions.

BSDS300 [Martin et al., 2001]	performance statistics
image size	240x160
computation time (ms)	1156 ± 185
mean fitting cost	2.51 ± 0.93
#surfaces	35.52 ± 8.91
PRI	0.74 ± 0.16

Table 2.3: This table gives an overview of the mean and standard deviation of the computation time, the mean fitting cost, the number of surfaces and the PRI, respectively.

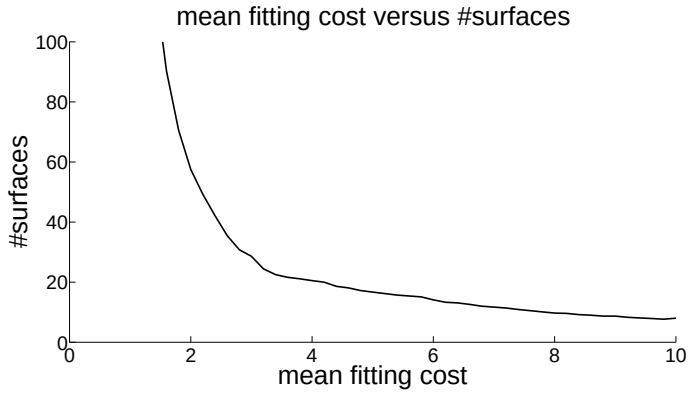


Figure 2.34: This graph plots the numbers of surface segments when segmenting images of the BSDS300 in function of mean fitting costs. For mean fitting costs of 3 or higher, the mean numbers of surface segments is relatively constant. For mean fitting costs of 3 or lower, the mean number of surface segments grows exponentially.

We compare the proposed segmentation algorithm based on a polynomial surface model, with segments produced by normalized cuts [Shi and Malik, 2000], mean shift [Comaniciu and Meer, 2002] and power watersheds [Couprie et al., 2011]. To demonstrate the difference between the proposed method and existing techniques, we perform three tests. The first test is performed on all images of the BSDS300. The remaining two tests are performed on two subsets of images of the BSDS300. The first subset includes highly textured images, while the second subset contains images of objects with diffuse reflecting surfaces. Examples for both subsets are the desert image and the face image in Figure 2.33, respectively. These subsets were carefully selected by computer vision experts.

A visual comparison of segmentation on a fish image with diffuse reflecting

PRI (higher is better)	BSDS300	Subset1	Subset2
Human	0.87	0.83	0.90
Polynomial surfaces	0.74	0.67	**0.84**
Power watersheds	0.77	0.74	0.79
Mean shift	0.76	0.72	0.77
Normalized cuts	0.72	0.69	0.74

Table 2.4: Comparing results in terms of the PRI when segmenting images of the BSDS300 with polynomial surfaces, power watersheds [Couprie et al., 2011], mean shift [Comaniciu and Meer, 2002] and normalized cuts [Shi and Malik, 2000], respectively. Three test are performed. One test on all images and two tests on subsets of images. The first subset includes textured images, while the second subset contains images of objects with diffuse reflecting surfaces. For this last subset, segmentation with polynomial surfaces significantly outperforms the other techniques.

surfaces is shown in Figure 2.35. Columns (b), (c), (d) and (e) on the second row in Figure 2.35 show segmentations of the fish image in Figure 2.35 (a), produced by normalized cuts, mean shift, power watersheds and the proposed segmentation technique based on a polynomial surface model, respectively. For these segmentation techniques, columns (f), (g), (h) and (i) on the third row show the segment approximations with polynomial surfaces of second degree as in Eq. (2.14). The power watershed algorithm uses the same seed pixels as produced by the proposed method. We can clearly ascertain that in contrast to the proposed method method, the segments produced by existing segmentation techniques do not always coincide with object features and the contours separating the surface segments often do not correspond to real image object edges. Furthermore, for existing techniques, the polynomial segment approximations do not accurately reconstruct the image.

An overview of the PRIs are given in Table 2.4. When testing the entire database, the segmentation results are comparable in terms of the PRI. As expected, the PRIs decrease when segmenting the more textured images of the first subset. For this subset, the proposed method performs less well when compared to the existing techniques. However, as for the desert image in Figure 2.33, the segmentation result is visually still acceptable. Next, when segmenting images of the second subset, segmentation with polynomial surfaces significantly outperforms the existing techniques. This confirms that our method is primarily aimed at segmenting images into flat, planar, convex, concave and saddle patches that correspond to meaningful parts of objects with diffuse reflecting surfaces. More comparative results on the BSDS300 in terms of the PRI are found in [Arbelaez et al., 2011].

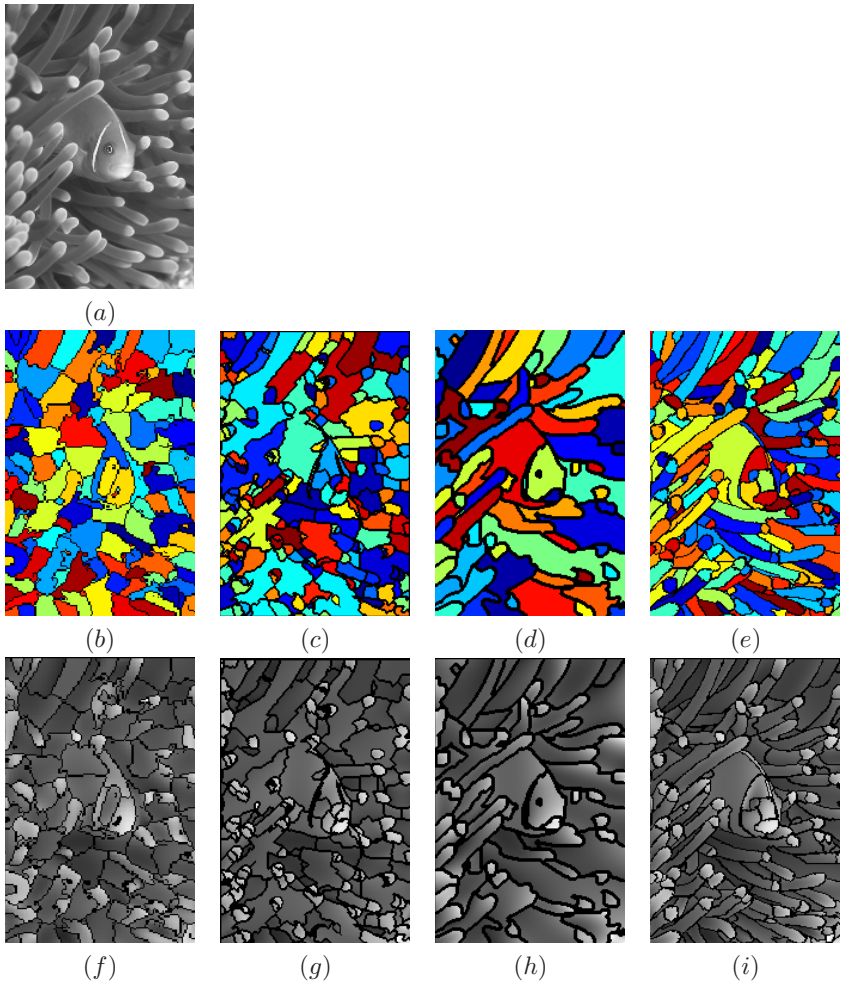


Figure 2.35: Visual comparison of image segmentation techniques on a fish image with diffuse reflecting surfaces. Columns (b), (c), (d) and (e) on the second row show segmentations of the fish image with diffuse reflecting surfaces in (a), produced by normalized cuts [Shi and Malik, 2000], mean shift [Comaniciu and Meer, 2002], power watersheds [Couprie et al., 2011] and polynomial surfaces, respectively. For these segmentation techniques, columns (f), (g), (h) and (i) on the third row show the segment approximations with polynomial surfaces of second degree as in Eq. (2.14). In contrast to existing segmentation techniques, the segments produced by the proposed method correspond more to object features, the contours separating the surface segments coincide with real image object edges and the low-degree polynomial approximations accurately reconstruct the image.

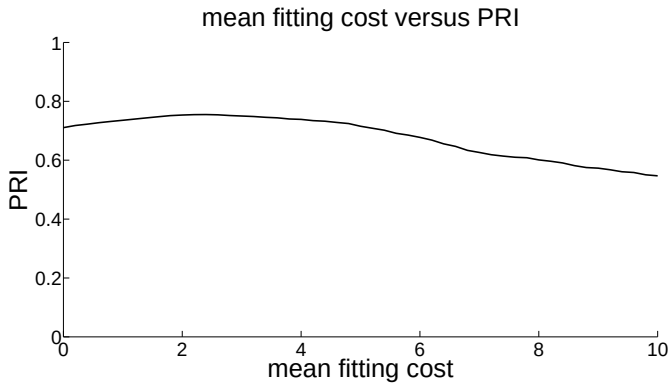


Figure 2.36: The graph in this Figure plots the mean fitting cost versus the PRI. The PRI is maximal for a mean fitting cost of 2.5.

As a last result, the graph in Figure 2.36 plots the mean fitting cost versus the PRI for testing the entire database. We conclude that for a mean fitting cost of 2.5 the PRI is maximal. As mentioned earlier, this maximum corresponds to the threshold parameters of adaptive region growing that are used in this work.

2.7 Conclusion

In this chapter, we presented novel 1-D and 2-D segmentation algorithms based on adaptive region growing and low-degree polynomial fitting to extract geometric low-level features from contour pixels and image intensities, respectively. These algorithms use a new adaptive thresholding technique with the L_∞ fitting cost as a segmentation criterion. The polynomial degree and the fitting error are automatically adapted during the region growing process. The main novelty is that the algorithms detect outliers, distinguish between strong and smooth discontinuities and find segments that are bent in a certain way, such as convex or concave segments. The polynomial curves and surfaces are grouped into two novel compact features: the Curve Edge Map (CEM) and the Surface Intensity Map (SIM), respectively. The polynomial functions approximate the pixels very closely. For faces, a representation with polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image.

This research was published in two papers [Deboeverie et al., 2013c, Deboeverie et al., 2013b] in international journals and one paper [Deboeverie et al., 2010] in the proceedings of an international conference. One journal publication was submitted [Deboeverie et al., 2013a].

3
Polynomial curve matching

3.1 Introduction

In this chapter, we propose algorithms for finding correspondences between low-level geometric features. Finding correspondences is still one of the fundamental problems in computer vision research, as many image processing applications require a solid and robust solution for matching problems, such as object recognition, object tracking and image registration. The purpose of a correspondence finding or matching algorithm is to indicate for a feature in one image which is the corresponding feature in a second image, where both image points must show the same 3D world point or shape.

Currently, the common approach for solving correspondence problems for static images consists of the following steps [Teelen, 2010]. (1) First a subset of characteristic features is extracted from both images by detecting remarkable patterns in the image intensity information. We assume that two sets of features (point features, curve features or region features) in two distinct images have been detected. In this work, we consider geometric low-level features, such as polynomial curves and surfaces. (2) The next step involves pairing the features from one image to their counterparts in the other image. The detected features in two distinct images can be matched by means of a (dis)similarity measure exploiting the image intensity values in their close neighbourhoods, the feature spatial distribution, or the feature symbolic description. Dissimilarity or distance is defined as a quantitative degree of how far apart two features are. A dissimilarity mea-

sure produces a higher value as two features become more distinct. Similarity or proximity quantifies the strength of relationship between two features. A similarity measure produces a higher value as two features become more similar. (3) The best match according to some similarity measure is not necessarily the correct match. One must additionally verify the spatial consistency of the pairs of similar features. Therefore, one must verify that each relation between a corresponding pair in both images follows the rules of some geometric transformation.

In this work, we propose methods to find correspondences in Curve Edge Maps (CEMs). The method to extract CEMs is described in Section 2.5. We introduce a dissimilarity function for local curve matching as well as a similarity function for global curve matching. The proposed functions match polynomial curves, based on shape, relative position and intensity. The main contribution is the introduction of intensity variations in the matching functions.

Representative applications for polynomial curve matching are object recognition and object tracking. Object recognition is the task of finding an object in an input image or video sequence from the images of objects in a database. Object tracking is the problem of identifying and following image elements moving across a video sequence automatically. In this thesis, we focus on the recognition and tracking of faces in images and video sequences. For object tracking purposes, in this chapter we propose a method to construct motion vectors for polynomial curve correspondence pairs. Furthermore, we will explain how to group the polynomial curve correspondence pairs in order to register the motion of objects.

The work in this chapter was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

This chapter is structured as follows: in Section 3.2, we discuss related work. In Section 2.3, we explain the basic principles of constructive polynomial fitting. In Section 2.4, we propose an adaptive region growing algorithm based on constructive polynomial fitting. This method is applied and evaluated for contour segmentation into polynomial curves and image segmentation into polynomial surfaces in Sections 2.5 and 2.6, respectively.

3.2 Related work

The relationship between features can be exploited introducing (dis)similarity functions for feature matching. Similarity measures are an essential ingredient in feature matching. Although the term similarity is often used, dissimilarity corresponds to the notion of distance: small distance means small dissimilarity, and large similarity. The algorithm to compute the (dis)similarity often depends on the precise measure, which depends on the required properties, which in turn depends on the particular matching problem for the application. The system can then either use the (dis)similarity measure to decide if the two features match or to form a

probability of the match. (Dis)similarity functions for matching often exploit the image intensity values in their close neighbourhoods, the feature spatial relationship, or other feature invariant descriptions.

Intensity-based methods include correlation-like methods [Pratt, 1991, Kaneko et al., 2003], Fourier methods [Bracewell, 1965, Loncaric, 1998] and mutual information methods [Viola and Wells, 1997, Rangarajan et al., 1999, Pluim et al., 2001]. Cross-correlation methods directly match image intensities without any structural analysis. Consequently, they are sensitive to intensity changes introduced by noise, varying illumination, ... and/or by using different sensor types. Fourier methods exploit the Fourier representation of the features in the frequency domain. Mutual information methods originate from the information theory. Mutual information is a measure of statistical dependency between two data sets and it is particularly suitable for matching of features from different modalities.

Methods using spatial relations exploit the information about the distance between features and their spatial distribution. A common technique is graph matching. Graphs are a general and powerful data structure for the representation of objects and concepts [Bunke, 2000]. A graph $G = (V, E)$ in its basic form is composed of vertices and edges. V is the set of vertices (also called nodes or points) and E is the set of edges (also known as arcs or lines) of graph G. In a graph representation, the nodes typically represent objects or parts of objects, while the edges describe relations between objects or object parts. Graphs have some interesting invariance properties. For instance, if a graph, which is drawn on paper, is translated, rotated, or transformed into its mirror image, it is still the same graph in the mathematical sense. These invariance properties, as well as the fact that graphs are well suited to model objects in terms of parts and their relations, make them very attractive for various applications.

In applications such as pattern recognition and computer vision, object similarity is an important issue. Given a database of known objects and a query, the task is to retrieve one or several objects from the database that are similar to the query. If graphs are used for object representation this problem turns into determining the similarity of graphs, which is generally referred to as graph matching. Standard concepts in graph matching include graph isomorphism, subgraph isomorphism, and maximum common subgraph. However, in real world applications we cannot always expect a perfect match between the input and one of the graphs in the database. Therefore, what is needed is an algorithm for error-tolerant matching, or equivalently, a method that computes a measure of similarity between two given graphs. For instance, Lades et al. [Lades et al., 1993] presented a dynamic link structure for distortion invariant object recognition which employs elastic graph matching to find the closest stored graph. Thus dynamic link architecture is an extension of classical artificial neural networks. Memorized objects are represented by sparse graphs, its vertices are labelled with a multiresolution description

in terms of a local power spectrum and the edges are labelled with geometrical distance vectors. Object recognition can be formulated as elastic graph matching, which is performed by stochastic optimization of a matching cost function. Wiskott and von der Malsburg [Wiskott and von der Malsburg, 1996] extended the technique. In general, the dynamic link architecture is superior to other face recognition techniques in terms of rotation invariant recognitions.

Methods using invariant descriptors estimate the correspondence of features can using their description, preferably invariant to the expected image deformation. The description should fulfill several conditions. The most important ones are invariance (the descriptions of the corresponding features from two distinct images have to be approximately the same), uniqueness (two different features should have different descriptions), stability (the description of a feature which is slightly deformed in an unknown manner should be close to the description of the original feature), and independence (if the feature description is a vector, its elements should be statistically independent). However, usually not all these conditions have to (or can) be satisfied simultaneously and it is necessary to find an appropriate trade-off. Features from two distinct images with the most similar invariant descriptions are paired as the corresponding ones. The choice of the type of the invariant description depends on the feature characteristics and the assumed geometric deformation of the images. Two well-known examples of invariant descriptors (scale invariance) are SIFT [Lowe, 2004], SURF [Bay et al., 2006, Bay et al., 2008], BRISK [Leutenegger et al., 2011] and FREAK [Alahi et al., 2012]. While searching for the best matching feature pairs in the space of feature descriptors, the minimum distance rule with thresholding is usually applied. An overview of several distance metrics, such as the Euclidean distance L_2, the City block distance L_1, the Chebyshev distance L_∞, the Bhattacharyya distance, the Chi-square distance χ^2, etc., is found in the work of Cha [Cha, 2007].

In this work, we propose a local and a global 1-d polynomial curve descriptor based on the image intensities, the shape and the spatial relationship. Closely related to our work, Takács [Takács, 1998] used edge maps to measure the similarity of face images. The faces were encoded into binary edge maps using the Sobel edge detection algorithm. The Hausdorff distance was chosen as a measure for the similarity of the two point sets, i.e., the edge maps of two faces, as it can be calculated without an explicit pairing of points in their respective data sets. The Hausdorff distance between point sets uses only the spatial information of an edge map without considering the inherent local structure and shape of the edges. Gao and Leung [Gao and Leung, 2002a] have successfully recognized faces by segmenting the edges into lines. The recognition system matches the lines of a Line Edge Map (LEM) of the query image with the LEM of the model image, using the Line Segment Hausdorff Distance. The LEM technique ensures that the sensitivity for noise and small changes in pose and expression is strongly reduced. LEM

achieves a good recognition rate under pose and illuminations variations, with one model per person. The performance can degrade abruptly, however, when the face is occluded, rotated or the facial expression differs strongly from the expression stored in the database.

3.3 Polynomial curve matching

In this section, we present two techniques to find correspondence pairs of polynomial curves in CEMs. The method to extract CEMs is described in Section 2.5. We introduce a dissimilarity function for local curve matching as well as a similarity function for global curve matching.

Local curve matching finds correspondence pairs of polynomial curves in consecutive images of a video sequence. A representative application is the tracking of a face or facial feature in a video sequence. Therefore, in Section 3.3.1, we present a dissimilarity measure for local curve matching which finds correspondence pairs of polynomial curves in aligned images with aligned objects. This work was published in [Deboeverie et al., 2008b]. This matching technique is very useful in applications where the main purpose is to track individual polynomial curves or groups of polynomial curves on a moving object in consecutive images of one video sequence, e.g. face tracking. Object tracking is the problem of identifying and following image elements moving across a video sequence automatically. Therefore, in this section we propose a method to construct motion vectors for polynomial curve correspondence pairs. Furthermore, we will explain how to group the polynomial curve correspondence pairs in order to register the motion of objects. This work was published in [Deboeverie et al., 2009b].

Global curve matching finds correspondence pairs of polynomial curves in database images and input images. A representative application is the recognition of a face of a person in an input image from the face images of persons in a database. Therefore, in Section 3.3.1, we present a similarity function for global curve matching which finds correspondence pairs of polynomial curves in non-aligned images with non-aligned objects. This work was published in [Deboeverie et al., 2011]. Here, the viewing orientation on the object is often slightly different. This matching technique is very useful in applications where the main goal is to recognize objects in two different images, e.g. face recognition.

3.3.1 Local matching

We introduce a polynomial curve distance measure that takes into account the distance and intensity differences along a polynomial curve. For example, each second degree polynomial curve defines a convex region, so that it makes sense to compute the difference between the average intensities near the boundary of

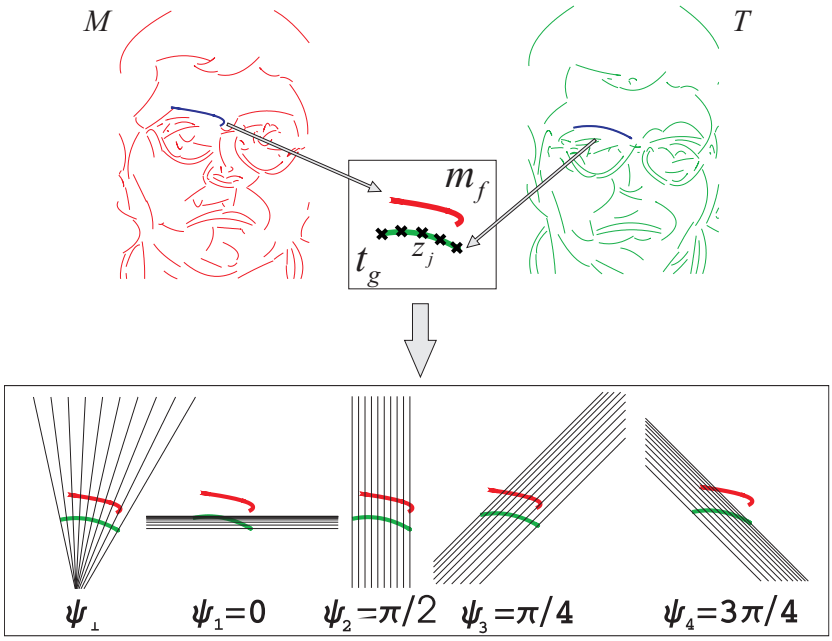


Figure 3.1: The matching process of polynomial curves. On top: two sets of polynomial curves M and T. Below: comparison of two polynomial curves m_f and t_g under different viewing orientations $\Psi = \{\psi_\perp\, \psi_1, \psi_2, \psi_3, \psi_4\}$.

this convex region. We will show that the matching rate improves considerably by measuring intensity variations.

Let $M = \{m_1, m_2, ..., m_f, ..., m_u\}$ and $T = \{t_1, t_2, ..., t_g, ..., t_v\}$ be two sets of low-degree (e.g. second degree) polynomial curves from different images, as shown on top in Figure 3.1. First, we define a distance measure $d(m_f, t_g)$ for matching the polynomial curve m_f to the polynomial curve t_g. We propose a combined measure of position, shape and intensity, which is largely independent from rotation, translation, scale, and global illumination. Although this is not essential for the definition of $d(m_f, t_g)$, to facilitate the computation of the distance measure, we subsample the polynomial curve m_f. We choose a collection of n equidistant viewing points $Z = \{z_1, z_2, \ldots, z_n\}$ on the parabolic segment m_f.

To obtain position, rotation and scale independency we compare m_f with t_g under different viewing angles, as shown in Figures 3.1. Let $\Psi = \{\psi_1, \psi_2, ..., \psi_r, \psi_\perp\}$ be a collection of $r + 1$ viewing orientations, where $\psi_1, \psi_2, ..., \psi_r$ denote r viewing angles and $\psi_\perp$ denotes a viewing angle perpendicular to the parabola m_f.

3.3.1.1 Shape distance measure

Let ψ_i be one of the viewing angles. The following steps are then executed for each ψ_i. We determine the s points of intersection $Q = \{q_1, q_2, ..., q_s\}$ of the straight lines with orientation ψ_i through the points z_j with the parabola segment t_g.That is, the intersection points are the points on t_g as seen from the points z_j in the direction ψ_i . In this work we use $n = 10$ viewing points and we require that $s \geq 5$, which means that the polynomial curves have to intersect with at least 50% of the lines. The process of building the distance measure as described below is illustrated for a vertical viewing orientation in Figure 3.2.

The average distance is

$$d_A(m_f, t_g) = \frac{1}{s}\sum_{k=1}^{s} d_k, \tag{3.1}$$

where d_k are the distances between the points of intersection $Q = \{q_1, q_2, ..., q_s\}$ and the viewing points $Z = \{z_1, z_2, ..., z_n\}$ measured along the viewing orientation. Let

$$\sigma_D^2(m_f, t_g) = \frac{1}{s}\sum_{k=1}^{s}(d_k - d_A)^2, \tag{3.2}$$

denote the variance of the distances, which we will use to measure shape dissimilarity between m_f and t_g.

The proposed distance measure also takes into account how far the segments have been shifted relative to each other. Therefore, we compute two distances d_{P_1} and d_{P_2} between the corresponding end-points of the polynomial curves and perpendicular on the viewing orientation, as shown in Figure 3.2. Then, the minimal distance measured perpendicular on the viewing orientation is

$$d_P(m_f, t_g) = \min(d_{P_1}, d_{P_2}). \tag{3.3}$$

We define

$$D_\Psi(m_f, t_g) = \sqrt{(d_A\sigma_{D_\psi})^2 + d_P^2}. \tag{3.4}$$

as a shape dissimilarity measure for two segments compared in the viewing direction ψ. By multiplying the average distance with the variance, we get a small value for $D_\Psi(m_f, t_g)$ when the polynomial curves are parallel and translated, and a large value otherwise.

3.3.1.2 Intensity distance measure

For the second, intensity dependent, part of the distance measure, we consider the intensities above and below the polynomial curve. One of the advantages of second degree polynomial curves over first degree polynomial curves is that second degree

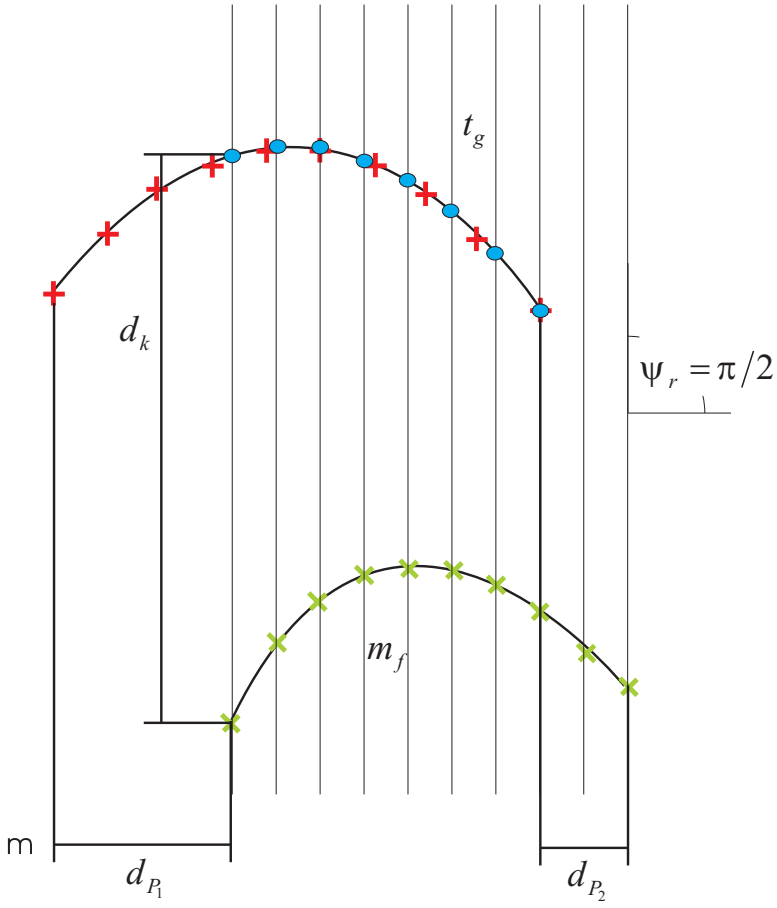


Figure 3.2: Local curve matching based on distance for a vertical viewing orientation. On the polynomial m_f there are selected ten viewing points. On the polynomial curve t_g eight points of intersection are shown. The distances are measured parallel with and perpendicular on the viewing orientation.

polynomial curves have a more clearly defined concave and convex side. We first introduce a dissimilarity measure for the intensity values at the convex side of the polynomial curve. The comparison of intensities as in the description below is illustrated for a vertical viewing orientation in Figure 3.3.

Let ψ be a viewing direction. For each viewing point z_j on the parabola segment m_f, we compute a weighted average intensity value $I_j^{conv}(m_f)$ at the convex

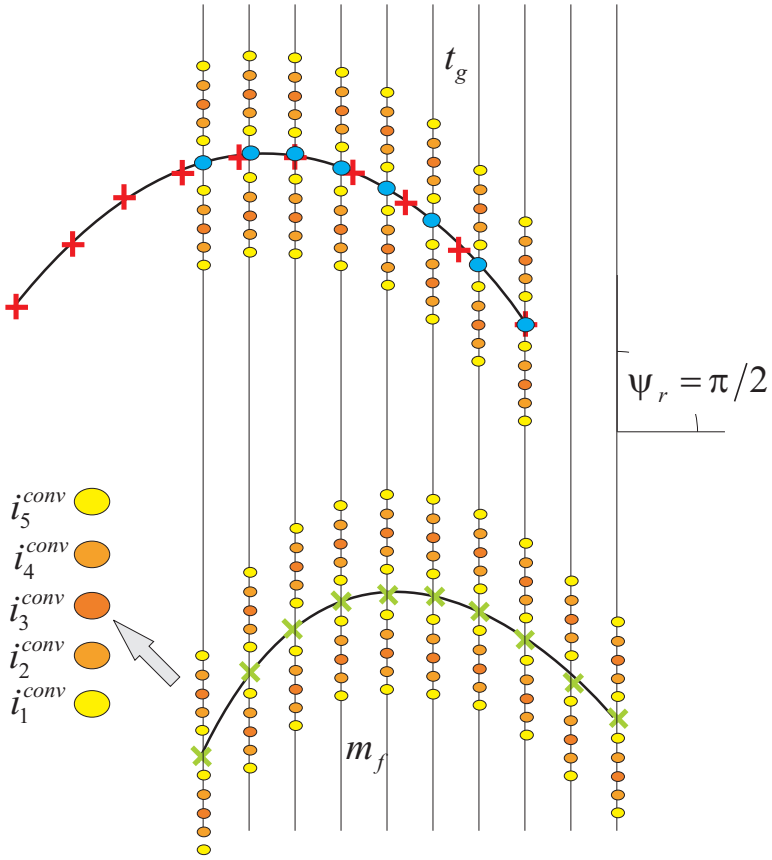


Figure 3.3: Local curve matching based on intensity for a vertical viewing orientation. On the parabola m_f ten viewing points are selected. On the parabola t_g eight points of intersection are shown. The pixels below and above the parabola segments which are taken into account for colour matching are shown as circles.

side,

$$I_j^{conv}(m_f) = \sum_{k=1}^{v} w_k i_k^{conv}(m_f) \text{ and } \sum_{k=1}^{v} w_k = 1, \tag{3.5}$$

where $i_k^{conv}(m_f)$ are the intensities (pixel values) at the convex side of the point z_j along the viewing direction. In this work we used $v = 5$. The intensities are weighted, because the pixels closest to the polynomial curves, and thus close to the edge map, can disturb the result. In this work we have used a Gaussian kernel for the weights w_k to increase the importance of pixels located centrally in the convex region.

Similarly, for each intersection point q_j on the polynomial curve t_g corresponding with z_j, we compute a weighted average intensity value $I_j^{conv}(t_g)$ at the convex side,

$$I_j^{conv}(t_g) = \sum_{k=1}^{v} w_k i_k^{conv}(t_g), \tag{3.6}$$

where $i_k^{conv}(t_g)$ are the intensities (pixel values) at the convex side of the point q_j.

The difference in average intensity at the convex sides of the parabola segments for the corresponding points z_j and q_j is

$$I_j^{conv}(m_f, t_g) = |I_j^{conv}(m_f) - I_j^{conv}(t_g)|. \tag{3.7}$$

The average difference in weighted average intensity at the convex sides of the polynomial curves is

$$I_A^{conv}(m_f, t_g) = \frac{1}{s} \sum_{k=1}^{s} I_k^{conv}(m_f, t_g). \tag{3.8}$$

The variance for the differences in intensity is

$$\sigma^2_{A_{conv}}(m_f, t_g) = \frac{1}{s} \sum_{k=1}^{s} (I_k^{conv}(m_f, t_g) - I_A^{conv}(m_f, t_g))^2. \tag{3.9}$$

Similarly, we define $I_A^{conc}(m_f, t_g)$ and $\sigma^2_{A_{conc}}(m_f, t_g)$ at the concave side.

The intensity dissimilarity function is defined as

$$I_\Psi(m_f, t_g) = \sqrt{(I_A^{conv}\sigma_{A_{conv}})^2 + (I_A^{conc}\sigma_{A_{conc}})^2}. \tag{3.10}$$

By multiplying the average differences in intensity with their variances, we reduce the value of the intensity measure when the polynomial curves have the same relative transition in intensity between their convex and concave side.

3.3.1.3 Matching cost

Finally, the disparity between the segment m_f and the segment t_g along the direction ψ is defined as

$$d_\psi(m_f, t_g) = \left(\frac{n}{s}\right)^2 \sqrt{D_\Psi(m_f, t_g)^2 + I_\Psi(m_f, t_g)^2}. \tag{3.11}$$

Scaling by $1/s^2$ reduces the disparity value for a higher number of intersection points.

The disparity between the segments m_f and t_g is defined as the minimum of the disparity over all viewing directions originating from m_f,

$$C(m_f, t_g) = \min_{\psi}(d_\psi(m_f, t_g)). \tag{3.12}$$

The polynomial curve m_f matches with the polynomial curve from the set T, for which the matching measure is minimal. The match is acceptable if this distance is below a certain threshold A (e.g. an experimentally defined value is $A = 10$):

$$C(m_f, T) = \min_{t_g \in T}(C(m_f, t_g)) \leq A. \tag{3.13}$$

Let $M' \subseteq M$ be the set of polynomial curves for which there are acceptable matches. The matching measure between the set of segments M and the set of segments T is defined as

$$S(M,T) = \frac{\sum\limits_{m_f \in M'} (l_{m_f} C(m_f, T))}{\sum\limits_{m_f \in M'} l_{m_f}} \left(\frac{|M| - |M'|}{|M|} \right), \tag{3.14}$$

where l_{m_f} is the length of the segment m_f, and $|M|$ and $|M'|$ represent the size of the two sets. Large polynomial curves have a larger weight than small segments. The measure for matching the segments M with segments T is made symmetrical by introducing the matching measure which will be used to compare faces:

$$H(M,T) = \max(S(M,T), S(T,M)). \tag{3.15}$$

An example of local curve matching in faces is shown in Figure 3.4. Corresponding second degree polynomial curves are indicated by the same colour. The graph in Figure 3.5 presents the result of local curve matching when testing an entire database [GTFD,]. The distribution on the left side describes the density of the matching cost for matching faces of the same person. The distribution on the right side describes the density of the matching costs for matching faces of different persons.

We evaluate the local matching technique by the application of people identification in Section 4.2.

3.3.1.4 Motion vectors

For tracking purposes, we construct motion vectors for polynomial curve correspondence pairs.

The tracking technique for the parabola segments in the consecutive frames is based on a matching method for individual parabola segments using both distance and intensity information as described in the previous Section 3.3.1. Since tracking is performed on consecutive frames, we introduce timing information in the notations. Let $M_0 = \{m_{01}, \ldots, m_{0f}, \ldots, m_{0u}\}$ and $M_1 = \{m_{11}, \ldots, m_{1g}, \ldots, m_{1v}\}$ be two sets of parabola segments from consecutive frames, where for each parabola segment the first index i is a time indicator, with each frame at $t-i$, and the second

Figure 3.4: Result of local curve matching in faces. The figure shows the second degree polynomial curves of two different faces of the same person. Corresponding polynomial curves are indicated by the same colour.

index is the index of the parabola segment in the set. The matching cost of two different parabola segments is a combined function of position, shape and intensity. To obtain position, rotation and scale independency, m_{0f} and m_{1g} are compared under $k+1$ different viewing angles $\psi_k = 0, \pi/k, 2\pi/k, \ldots, (k-1)\pi/k$ and $\psi_\perp$.

For each viewing angle ψ_k, the shape dissimilarity function D_{ψ_k} is

$$D_{\psi_k}(m_{0f}, m_{1g}) = \sqrt{(d_k^{par}\sigma_k^{par})^2 + (d_k^{per})^2}, \tag{3.16}$$

where d_k^{par} is the average of the distances measured parallel along the viewing orientation ψ_k, σ_k^{par} denotes the variance of the distances, and d_k^{per} is the minimum distance measured perpendicular to the viewing orientation.

The intensity dissimilarity function I_{ψ_k} in the viewing orientation ψ_k, is defined as

$$I_{\psi_k}(m_{0f}, m_{1g}) = \sqrt{(i_k^{cv}\sigma_k^{cv})^2 + (i_k^{cc}\sigma_k^{cc})^2}, \tag{3.17}$$

where i_k^{cv} and σ_k^{cv} are the average differences in average intensity and the variance for the differences in intensity at the convex sides of the parabola segments, respectively. i_k^{cc} and σ_k^{cc} are defined similarly at the concave side.

We search for a one-to-one match for each of the parabola segments in the current frame by a minimization of the dissimilarities D_{ψ_k} and I_{ψ_k} between two

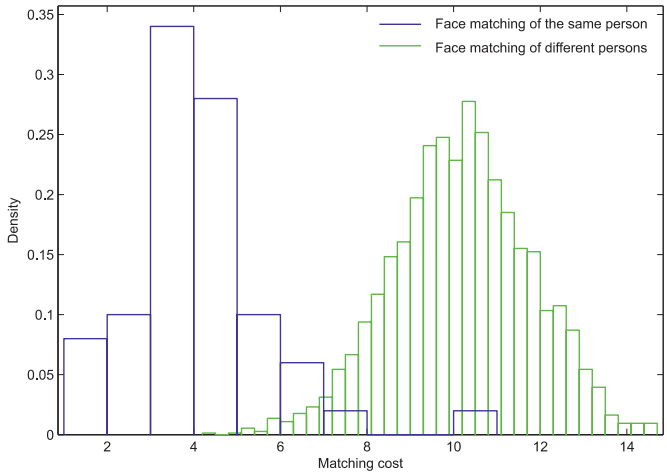


Figure 3.5: Result of local curve matching in faces when testing an entire database [GTFD,]. The distribution on the left side describes the density of the matching cost for matching faces of the same person. The distribution on the right side describes the density of the matching costs for matching faces of different persons.

segments m_{0f} and m_{1g} over all ψ_k. An example of matching parabola segments in two consecutive frames of a vehicle is shown in figure 3.6, corresponding parabola segments are indicated by the same colour.

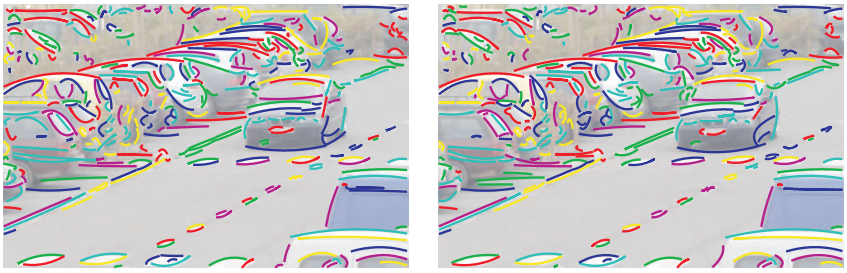
The motion vector $\overrightarrow{v_{0fg}}\{p_{0fg}, \Delta x_{0fg}, \Delta y_{0fg}\}$, at $t = 0$ for the unique parabola correspondence pair $\{m_{0f}, m_{1g}\}$ is then defined by three parameters,

- a location indicator $p_{0fg} = c_f$, with c_f the center point of the segment m_{0f};
- a movement along the x-axis $\Delta x_{0fg} = d_k^{par} \cos \psi_k$;
- and a movement along the y-axis $\Delta y_{0fg} = d_k^{par} \sin \psi_k$;

3.3.1.5 Motion registration

Now the individual motion vectors for the polynomial curve correspondence pairs must be grouped so that we can define the motion of the objects composed of clusters of polynomial curves.

For every polynomial curve m_{tf} in the current frame $t = 0$, we construct a chain $C_f = \{(m_{0f}, m_{1g}), (m_{1g}, m_{2h}), \ldots, \}$ of polynomial curve correspondences over maximum the last Q_m frames, i.e. we look in each previous frame for the single best matching polynomial curve. If there is no correspondence in a



(a) (b)

Figure 3.6: Result of local curve matching in vehicles. (a): The second degree polynomial curves of the same vehicle in two consecutive frames and the matching results, corresponding polynomial curves are indicated by the same colour.

previous frame, the chain is broken. For each chain, we collect the motion vectors $\overrightarrow{v_{tfg}}$ for all $Q < Q_m$ correspondence pairs in the chain, with $t = 0, 1, \ldots, Q-1$. We denote this set of motion vectors by S_f for the polynomial curve chain C_f. S_f is characterized by three parameters $\{p_{S_f}, \Delta x_{S_f}, \Delta y_{S_f}\}$:

- a location parameter $p_{S_f} = p_{0fg}$;
- an average movement Δx_{S_f} along the x-axis over the last Q frames, i.e.

$$\Delta x_{S_f} = \frac{1}{Q} \sum_{t=0}^{Q-1} \Delta x_{tfg}; \tag{3.18}$$

- and an average movement Δy_{S_f} along the y-axis over the last Q frames,

$$\Delta y_{S_f} = \frac{1}{Q} \sum_{t=0}^{Q-1} \Delta y_{tfg}. \tag{3.19}$$

Among the advantages of using the chain of polynomial curves are, first, a smoothing effect on the trajectory of the object, i.e. of the individual polynomial curve clusters, second, longer chains get more weight in the computations of the cluster parameters, and third, we could introduce a learning parameter for a more advanced foreground/background segmentation algorithm.

In a first step we do foreground/background segmentation as follows: we verify whether the polynomial curve m_{0f} has actually moved during the last Q frames. Therefore we check whether its average distance $\sqrt{(\Delta x_{S_f})^2 + (\Delta y_{S_f})^2}$ is higher than a preset threshold T. In our experiments, we choose $T = 0.5$, so that we only take those parabola segments into account which are sufficiently moving.

Otherwise, the polynomial curve is assigned to the set of background polynomial curves of the current scene.

In the second step we cluster the moving polynomial curves into individually moving objects. The criteria for clustering are:

- the segments are in each others neighbourhood, temporally and spatially,
- the segments are moving with the same velocity,
- and in the same direction.

Initially, there are no clusters, so the first polynomial curve chain C_1 defines a new cluster ω_1 in the cluster set Ω. The cluster ω_1 has center point $p_{\omega_1} = p_{S_1}$, an average movement along the x-axis $\Delta x_{\omega_1} = \Delta x_{S_1}$ and an average movement along the y-axis $\Delta y_{\omega_1} = \Delta y_{S_1}$, i.e. it has a cluster motion vector $\overrightarrow{v_{\omega_1}}(p_{\omega_1}, \Delta x_{\omega_1}, \Delta y_{\omega_1})$.

For each new chain C_f, we verify whether it belongs to an existing cluster ω_n from the set of clusters $\Omega = \omega_1, \ldots, \omega_j, \ldots, \omega_n$. Otherwise, C_f defines a new cluster ω_{n+1}. C_f belongs to a cluster ω_j when it satisfies three conditions.

1. The Euclidean distance R_{fj} from p_{S_f} to the current cluster center point p_{ω_j} must be below the user specified radius R.

2. The polynomial curves in a cluster must move with the same velocity. When including C_f with movements Δx_{S_f} and Δy_{S_f}, we can compute the new movement along the x-axis as

$$\Delta x_{\omega_j}^{new} = \frac{m \Delta x_{\omega_j}^{old} + \Delta x_{S_f}}{m+1}. \tag{3.20}$$

   with m the number of parabola segment chains in the cluster ω_n. Similarly, we compute $\Delta y_{\omega_j}^{new}$, the new movement along the y-axis.

   After inclusion, the variance σ_{lj}^2 of the lengths for all S_r in the cluster ω_j must be below V_l, i.e.

$$\sigma_{lj}^2 = \frac{1}{m+1} \sum_{r=1}^{m+1} (\sqrt{(\Delta x_{S_r})^2 + (\Delta y_{S_r})^2} - \sqrt{(\Delta x_{\omega_j}^{new})^2 + (\Delta y_{\omega_j}^{new})^2})^2 < V_l. \tag{3.21}$$

3. The polynomial curves in one cluster must move in the same direction. Therefore, the variance $\sigma_{\gamma j}^2$ of all directions for all S_r in the cluster ω_j must be below V_γ, i.e.,

$$\sigma_{\gamma j}^2 = \frac{1}{m+1} \sum_{r=1}^{m+1} (\tan^{-1} \frac{\Delta y_{S_r}}{\Delta x_{S_r}} - \tan^{-1} \frac{\Delta y_{\omega_j}^{new}}{\Delta x_{\omega_j}^{new}})^2 < V_\gamma. \tag{3.22}$$

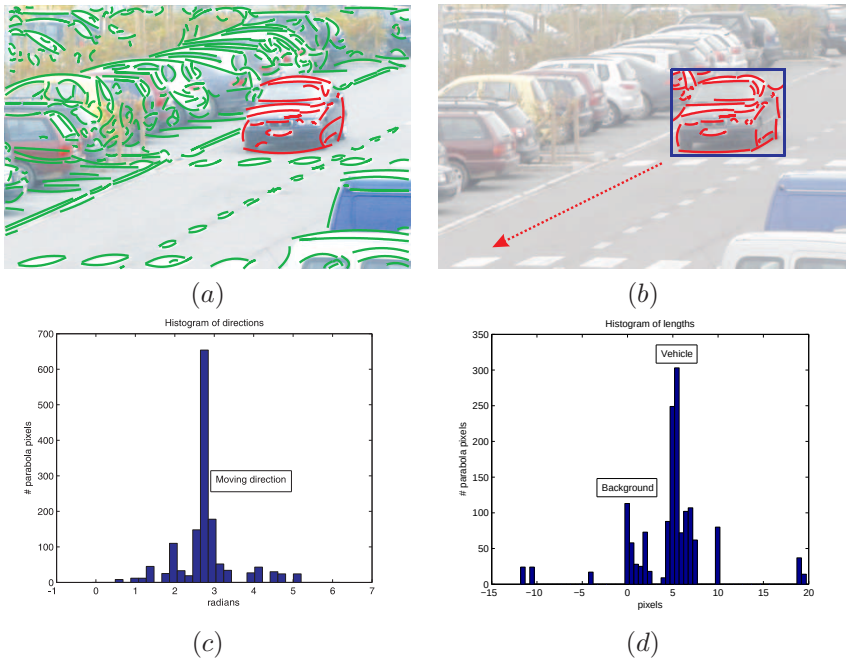


Figure 3.7: (a): A cluster of moving polynomial curves, indicated by the colour red. The background polynomial curves are indicated by the colour green. (b): The minimal enclosing bounding box. (c) and (d): The histograms of directions and lengths from the polynomial curve chains in the red cluster, respectively.

When C_f satisfies the requirements for more than one cluster, the combination $aR_{fj} + b\sigma_{lj}^2 + c\sigma_{\gamma j}^2$ is minimized. When C_f is added to the cluster ω_j, we update its motion vector $\overrightarrow{v_{\omega_j}}$, i.e. its center point and its average movements along the x- and y-axis as defined above.

Figure 3.7 (a) shows an example of a detected cluster of moving polynomial curves. The polynomial curves are indicated by the colour red, while the background polynomial curves are green. In Figure 3.7 (b) the minimal enclosing bounding box is defined for a cluster of polynomial curves.

The main direction in which the vehicle moves can also be detected in the histogram in Figure 3.7 (c), in which the directions for all S_r in the bounding box of Figure 3.7 (b) is shown. There are 32 bins in the histogram, so the width of each bin is $\frac{\pi}{16}$. The average direction of the cluster corresponds to the position of the peak in the histogram.

The cluster translation distance can also be estimated from the histogram of the lengths for all S_r in the bounding box, projected perpendicular on the axis

of the main direction. The width of the bins is 0.5 pixels in the histogram of Figure 3.7 (d). The first blob is caused by background polynomial curves, due to the representation of the rigid object by a bounding box, i.e. there are also some background polynomial curves included.

We will evaluate motion registration by the application of vehicle tracking in the next chapter in Section 5.2.

3.3.2 Global matching

In this section, we employ global matching of polynomial curves in different CEMs to find corresponding polynomial curves. For global matching, the polynomial curves do not have to be in each others neighbourhood. The matching technique we propose here does not require that the CEMs are aligned. It considers two characteristics of the polynomial curves.

- The first characteristic is the intensity difference between the inner and outer side of the polynomial curve. Each polynomial curve defines a convex region, so that it makes sense to distinguish polynomial curves from facial features with intensity histograms between the inner and outer side.
- The second characteristic is the relative position of the polynomial curve in the face CEM, found by the characteristic positions of the the polynomial curves from the facial features in the face CEMs. This topology is modelled by a histogram of relative positions in log-polar space, which is based on the work of shape contexts from Belongie et. al. [Belongie et al., 2002].

The two characteristics classify an individual polynomial curve in the face CEM, which is useful for facial feature classification.

3.3.2.1 Intensity histograms

One of the discriminative properties in our system is the difference in intensity between the inner and outer side of the polynomial curve. For each side we construct a normalized intensity histogram.

When estimating a histogram from one side of the polynomial curve, the region of interest for intensities is in the area between the original polynomial and a duplicate which is translated parallel to the main axis of the polynomial curve.

Figure 3.8 shows on the left two intensity histograms from the inner and outer side from the polynomial curve of the left eyebrow for a face of the Bern University Face Database [Bern,]. The histogram representing the upper side has its intensities on the bright side, while the histogram representing the lower side has its intensities on the dark side.

To match the intensity histograms from one side of two different polynomial curves, we use the Bhattacharyya distance metric or B-distance measure, which

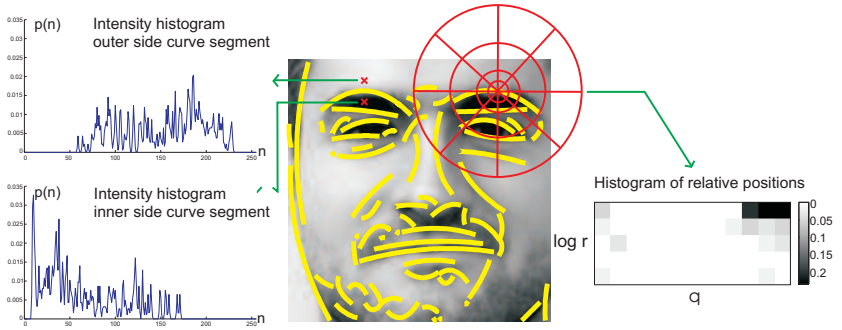


Figure 3.8: Histograms of intensities and relative positions. On the left are shown the intensity histograms from the inner and outer side of the polynomial curve, which describes the left eyebrow. On the right is plotted the log-polar histogram of relative positions of the polynomial curve, which describes the right eyebrow.

measures the similarity of two probability distributions and is a value between 0 and 1. This measure is chosen for its good classification properties [Cha and Srihari, 2002]. The B-distance between two intensity histograms f and g is

$$B(f,g) = 1 - \sum_{l=0}^{L} \sqrt{f(l)g(l)}, \tag{3.23}$$

where L is the number of bins in the histograms, $L = 255$ for gray images. The matching of intensity histograms is done for the inner and outer side of the polynomial curves, resulting in B_{in} and B_{out}.

3.3.2.2 Histograms of relative positions

A polynomial curve in the CEM is characterized by its relative position. The relative position becomes important when classifying individual facial features, for example to distinguish between polynomial curves from the eyebrow and the upper lip, which have similar transitions in intensities between the inner and the outer side. In our work, the topology of the face CEM is modelled using shape contexts. In the original shape context approach [Belongie et al., 2002], a shape is represented by a discrete set of sampled points $P = p_1, p_2, \ldots, p_n$. For each point $p_i \in P$, a coarse histogram h_i is computed to define the local shape context of p_i. To ensure that the local descriptor is sensitive to nearby points, the local histogram is computed in a log-polar space. The best matching results are obtained with a histogram computed for the center point of the polynomial curve in the face CEM. When considering such a center point, the sampled points P are the discretized points on the other polynomial curves in the face CEM. An example of a histogram

of relative positions is shown in Figure 3.8, on the right is plotted the log-polar histogram of the polynomial curve of the right eyebrow. In practice, the circle template covers the entire face.

Assume that p_i and q_j are the center points of the polynomial curves of two different faces. The shape context approach defines the cost of matching the two polynomial curves by the following χ^2 test statistic:

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \tag{3.24}$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histograms of relative positions of p_i and q_j, respectively.

3.3.2.3 Matching cost

We look for a one-to-one match for each of the polynomial curves by a minimization of the linearly combined cost $D_m = a(B_{in} + B_{out})/2 + bC$, for a pair m of matching polynomial curves, where B_{in} and B_{out} are costs for matching intensity histograms and C is the cost for matching histograms of relative positions. From the pairs of matching polynomial curves, a global length weighted matching cost E is computed, with

$$E = \sum_{m=1}^{M} D_m l_m / \sum_{m=1}^{M} l_m, \tag{3.25}$$

where M is the number of unique pairs of polynomial curves and l_m is the average length of the matching polynomial curves pair m. Then, a face is recognized when the global matching cost reaches a minimum.

We will evaluate the global matching technique by the applications of people identification and best view selection in the next chapter in Sections 4.2 and 4.3, respectively.

3.4 Conclusion

In this chapter, polynomial curve correspondence pairs are found by a technique that matches polynomial curves from different faces, based on distance and intensity. We make a distinction between a local and a global matching technique. The difference lies in the application: local matching is especially used in tracking applications, while global matching focuses on recognition applications.

This research was published in four papers [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011] in the proceedings of international conferences.

4

Face analysis applications

4.1 Introduction

Automatic, reliable and fast face analysis is becoming an area of growing interest in computer vision research. This chapter presents an automatic and real-time system for face analysis, which includes recognition and tracking of faces and facial features, and may be used in visual communication applications, such as video conferencing, virtual reality, human machine interaction, surveillance, etc.

In this chapter, we evaluate the matching techniques for maps of polynomial curves (CEMs) that were introduced in the previous chapters for recognition and tracking of faces. The method to extract CEMs is described in Section 2.5. A local and a global matching method for CEMs are described in Sections 3.3.1 and 3.3.2, respectively. We use a top-down approach. Firstly, faces are recognized and tracked. Then, individual facial features are classified and tracked. The applications considered are people identification, best view selection and behaviour analysis applications, such as entering/leaving detection, head movement detection and speaker detection. These applications are extensively evaluated on a large number of representative databases and video sequences. Furthermore, our methods are compared to several techniques of the state of the art. For instance, we make a comparison with a technique that uses line features and achieve better results in experiments on different databases.

The work in this chapter was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2011, Deboeverie et al., 2012].

This chapter is structured as follows: people identification and best view selection are treated in Sections 4.2 and 4.3, respectively. Section 4.4 treats behaviour analysis applications, such as entering/leaving detection, head movement detection and speaker detection in Sections 4.4.1, 4.4.2 and 4.4.3, respectively.

4.2 People identification

Although face recognition systems are already used in real-life applications, such as identification with bank cards, access control, security control and supervision systems, reliable and fast recognition of faces in various circumstances is still a challenging scientific problem. Current research is aimed at further improving the reliability, efficiency and applicability of face recognition algorithms. Faces are similar in structure and show only small differences from person to person. Aspects like differences in lighting, the large variety of distinct face expressions, the orientation and relative position of the face, changes in hairstyle or the presence of glasses make the recognition still more complicated. The difference in appearance of a single face due to pose, expression and illumination are often larger than the differences between two faces of two different persons under similar circumstances. In addition, efficient coding is needed to reduce the size of face model databases, which should contain a single description for each face [Martinez, 2002, Kim and Kittler, 2005]. Another desirable property, becoming prominent in recent applications, is that the recognition and coding algorithms can also be used to solve more general problems such as the recognition of pose and face expression, occluded or imprecisely localized faces, gender recognition or age estimation [Cevikalp et al., 2005].

4.2.1 Related work on people identification

In this work, we model the face as a flexible ellipsoid mask with cutouts for the eyes, the mouth, the nose and the nostrils. The contour pixels and the image intensities of the different facial parts are represented by polynomial surfaces and curves that are convex or concave. The flexibility of the model is obtained by allowing polynomials with a variable degree and a variable approximation error. Our model is inspired by different face models proposed in literature. Yuille et al. [Yuille et al., 1992] proposed deformable templates based on simple geometrical shapes that can deform and move for locating eyes and mouths. Brunelli et al. [Brunelli and Poggio, 1993] introduced two models, the first one with geometrical features, such as nose width and length, mouth position and chin shape, and the second one based on grey-level template matching. Lanitis et al. [Lanitis et al., 1995, Lanitis et al., 1997] proposed a flexible model that represents both shape and grey-level appearance in a point distribution model. Xu et al. [Xu et al., 2008] proposed a

hierarchical model of faces as a three-layer graph to take into account structural variabilities over multiple resolutions. The first layer treats each face as a whole, the second layer refines the local facial parts jointly as a set of individual templates, and the third layer further divides the face into 15 zones and models facial features such as eye corners, marks, or wrinkles. Ding et al. [Ding and Martinez, 2010] introduced a model based on texture patterns. To improve the detection of internal facial features, i.e., eyes, brows, nose, and mouth, they added context information of each facial feature correlated with the surroundings. Different from these methods is that we allow local as well as global flexibility in our model by adaptive sampling of the face region with constructive fitting. In this chapter, we apply our face model to several face analysis applications, such as face recognition.

There exist a large number of face recognition algorithms, from utilizing the facial properties and relations, such as areas, distances, and angles [Cox et al., 1996] to projecting face image to feature spaces, such as Eigenface [Turk and Pentland, 1991], Fisherface [Belhumeur et al., 1997], Laplacianface [He et al., 2005] and derivative domain [Kim et al., 2005, Zhao et al., 2003]. However, those methods were designed for well aligned, uniformly illuminated, and frontal face images. In practice, it is almost impossible to satisfy these requirements, especially in security surveillance system. Consequently, many efforts have been made to develop algorithms for unconstrained face images [Wright and Hua, 2009, Dreuw et al., 2009, Wolf et al., 2008, Ruiz-del Solar et al., 2009]. Instead of using global features, they advocated using local appearance descriptors such as Gabor jets [Zou et al., 2007, Tan and Triggs, 2007], SURF [Bay et al., 2006], SIFT [Lowe, 2004], HOG [Albiol et al., 2008] and Local Binary Patterns [Ojala et al., 2000]. Local appearance descriptors are more robust to occlusion, expression and small sample sizes than global features.

Traditionally, face recognition algorithms are classified as being either holistic or feature based. Holistic techniques based on Principal Component Analysis (PCA), can obtain high recognition rates, but are sensitive to variations in pose and facial expression. Techniques based on Linear Discriminant Analysis (LDA) [Kim et al., 2005] try to cope with the shortcomings of the techniques based on PCA. These approaches, however, still require a large database of faces with various poses and expressions for training. Martinez uses local regions that become more robust for variation of expression and occlusion [Martinez, 2002]. Multimodal algorithms try to combine local and global analysis techniques to improve robustness and reliability [Mian et al., 2007].

Facial analysis has generally been addressed by algorithms that use models based on shape and texture. The Active Shape Model (ASM) proposed by Cootes et. al. [Cootes et al., 1992] is one of the early approaches that attempts to fit the data with a model that can deform in ways consistent with a training set. The Active Appearance Model (AAM) [Cootes et al., 2001] is a popular extension of the

ASM. AAM is an integrated statistical model which combines a model of shape variation with a model of the appearance variations in a shape-normalized frame. The recently proposed Boosted Appearance Model (BAM), proposed by Liu et. al. [Liu, 2007,Liu, 2009], uses a shape representation similar to AAM, whereas the appearance is given by a set of discriminative features, trained to form a boosted classifier, able to distinguish between correct and incorrect face alignment. Liang et. al. [Liang et al., 2008] proposed a component-based discriminative approach for face alignment without requiring initialization. A set of learned direction classifiers guide the search of the configurations of facial components among multiple detected modes of facial components. In Elastic Bunch Graph Matching (EBGM), proposed by [Wiskott and von der Malsburg, 1996], all human faces share a similar topological structure. Faces are represented as graphs, with nodes positioned at fiducial points and edges labelled with 2-D distance vectors. Each node contains a set of 40 complex Gabor wavelet coefficients at different scales and orientations (phase, amplitude). They are called jets. Recognition is based on labelled graphs, which are sets of nodes connected by edges, nodes are labelled with jets, edges are labelled with distances.

In this work, faces are represented with Curve Edge Maps, which are collections of polynomial segments with a convex region. As explained in Chapter 2, the segments are extracted from edge pixels using an adaptive incremental linear-time fitting algorithm based on constructive polynomial fitting. As explained in Chapter 3, we find correspondences in CEMs by a technique that matches polynomial curves, based on shape, relative position and intensity. The face analysis system in this work has the advantages of simplicity, real-time performance and extensibility to the different aspects of face analysis.

4.2.2 Method of people identification

In this work, we perform face recognition for each person in a database by comparing the input face model to the face models of all other persons in the database. We consider face CEM as face model. The method to extract face CEM is thoroughly explained in Section 2.5. Then, a person is recognized correctly if the matching cost between its input face model and the person's own face model in the database is a minimum and below a predefined threshold. We consider the matching cost (Eq. 3.15) as defined for local matching of face CEMs in Section 3.3.1, as well as the matching cost (Eq. 3.25) for global matching of face CEMs, as defined in Section 3.3.2. The method overview of people identification is also represented in a block diagram in Figure 4.1. Note that input face images in video sequences are firstly detected using the cascade-based face detection algorithm of Viola and Jones [Viola and Jones, 2001].

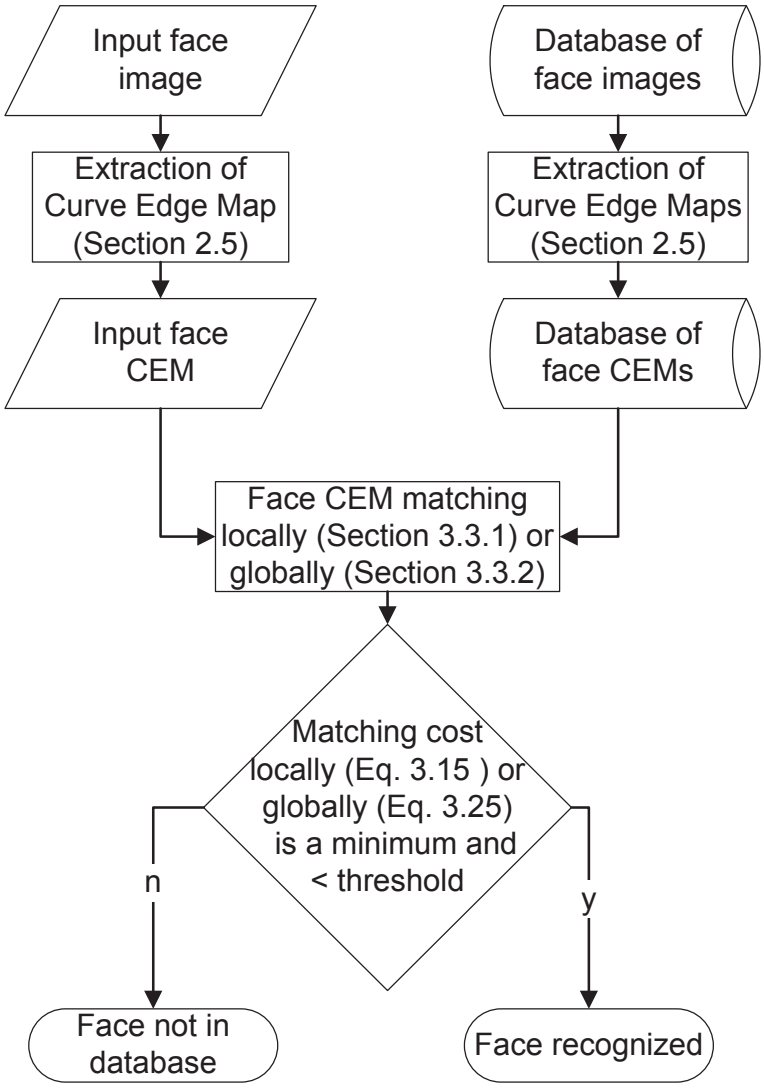


Figure 4.1: This block diagram represents the method overview of people identification. We perform face recognition for each person in a database by comparing the input face model to the face models of all other persons in the database. We consider face CEM as face model. Then, a person is recognized correctly if the matching cost of its input face model and the person's own face model in the database is a minimum and below a predefined threshold.

We evaluate the local and global matching of face CEMs for all conditions of human face recognition, i.e., face recognition under controlled/ideal condition, varying lighting condition, varying facial expression, and varying pose. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2011]. Here, the CEMs consist of collections of second-degree polynomial curves or parabolas. The system performances are compared to the LEM method [Gao and Leung, 2002b].

Our system applications are evaluated on different face databases and video sequences, considering variations in scale, lighting, facial expressions and pose. The publicly available databases considered are the Georgia Tech Face Database (GTFD) [GTFD,], the Database of Faces (ATT) [ATT,], the Bern University Face Database (BERN) [Bern,], the AR Face Database (AR) [Martinez and Benavente, 1998], the Yale University Face Database (YFD) [Yale,] and the BioID face database (BioID) with ground truth marker points [Jesorsky et al., 2001].

4.2.3 Evaluation of people identification

Local curve matching

For each person in a database we compare a face input to the face models of all other persons. A person is recognized correctly if the matching cost (Eq. 3.15) of its face input and the person's own face model is a minimum. An example of face CEM matching obtained from the method in 3.3.1 is shown in Figure 4.2(a), corresponding parabolas are indicated by the same colour. The result of the LEM method is shown in Figure 4.2(b). The graphs in Figures 4.3 (a) and (b) show the face recognition results of matching CEM and LEM, respectively, when testing an entire database. The distributions at the left side of each graph describe the density of the matching cost for matching faces of the same person. The distributions on the right side of each graph describe the density of the matching costs for matching faces of different persons. For CEM the distributions are more separated, which indicates that the CEM method performs better than the LEM method.

Recognition results on all databases are shown in Table 4.1. The second and the third column show the results for the method based on PCA [Turk and Pentland, 1991] and the LEM method, respectively. The fourth and the fifth column show the results for CEM matching using the distance measure and the intensity measure, respectively. The sixth column shows the results using a combination of the distance and the intensity measure. The results are top-1 classification, the correct match is only counted when the best matched face from a model is the correct person.

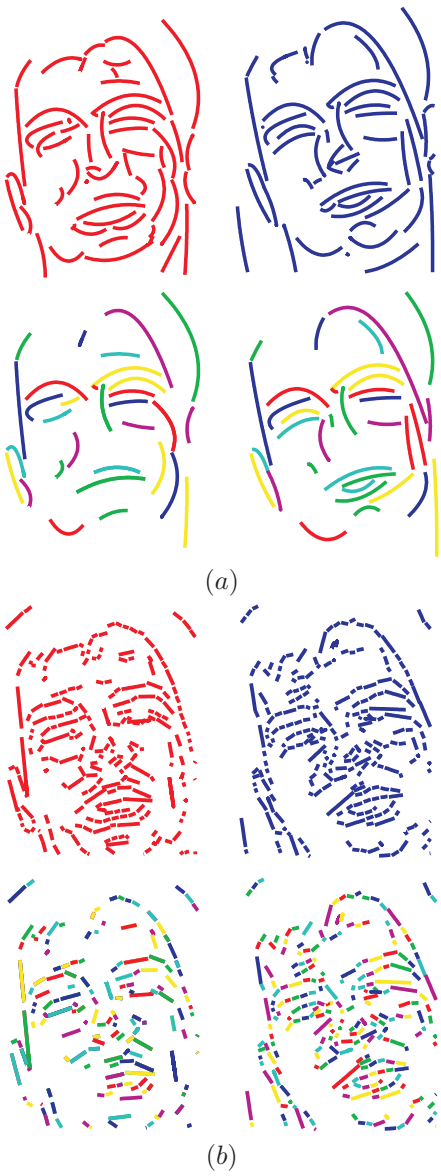
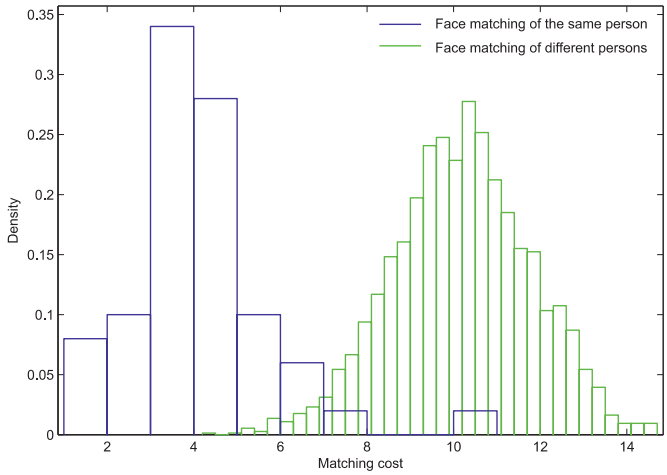
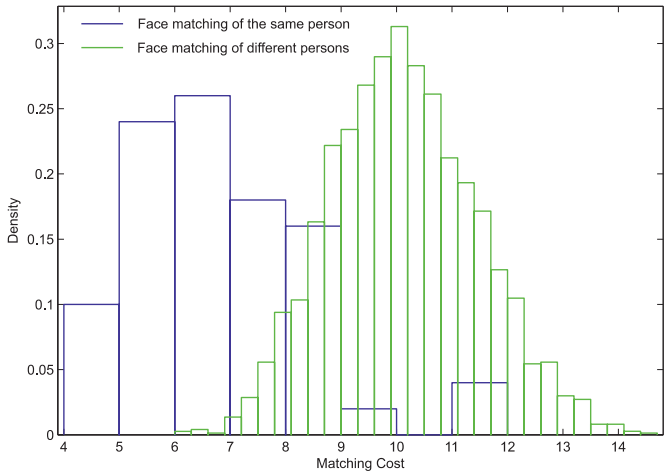


Figure 4.2: Result of face recognition using local curve matching. (a): The second-degree polynomial curves or parabola segments (CEMs) of two different faces of the same person. Corresponding parabola segments found are indicated by the same colour. (b): The corresponding line segments (LEM) of two different faces of the same person.



(a)



(b)

Figure 4.3: The graphs in (a) and (b) indicate the face recognition results of matching CEM and LEM, respectively, when testing an entire database [GTFD,]. The distributions on the left side of each graph describe the density of the matching cost for matching faces of the same person. The distributions on the right side of each graph describe the density of the matching costs for matching faces of different persons. For CEM the distributions are more separated, which indicates that the CEM method performs better than the LEM method.

	PCA based	LEM	CEM Distance	CEM Intensity	CEM Combination
Controlled conditions					
GTFD	86,00%	84,00%	72,00%	96,00%	**98,00%**
ATT	94,00%	95,00%	90,00%	97,50%	**100,00%**
BERN	86,66%	80,00%	93,33%	100,00%	**100,00%**
AR	96,60%	96,40%	88,10 %	100,00%	**98%**
Pose variation					
BERN Right	57,14%	55,00%	68,34%	90,00%	**93,34%**
BERN Left	51,87%	48,33%	68,34%	91,67%	**86,67%**
BERN Up	44,12%	46,67%	70,00%	88,33%	**86,67%**
BERN Down	44,03%	45,00%	68,34%	78,34%	**73,34%**
Size variation					
AR with size variation	55,69%	53,80%	70,56%	85,30%	**90,21%**
Lighting condition variation					
AR with left light on	70,32%	92,86%	80,12%	93,27%	**96,34%**
AR with right light on	71,73%	91,07%	82,10%	94,40%	**94,84%**
AR with both lights on	68,56%	74,11%	78,30%	88,67%	**92,10%**
Facial expression variation					
AR with smiling expr.	74,92%	78,57%	85,26%	98,34%	**96,53%**
AR with angry expr.	77,56%	92,86%	82,30%	96,11%	**97,30%**
AR with screaming expr.	59,06%	31,25%	71,62%	98,34%	**96,21%**

Table 4.1: Face recognition results using local curve matching: % correctly recognized faces. We test face recognition under controlled condition and size variation, under varying lighting condition, under varying facial expression and under varying pose. The proposed face recognition technique performs consistently superior to (or equally well as) the LEM method and the method based on Eigenfaces (PCA) in all comparison experiments. In general, CEM combination can achieve the best performance level. However, in some cases, CEM intensity outperforms CEM combination: this is when the relative rotation of the faces is too large.

We test face recognition under controlled condition and size variation, under varying lighting condition, under varying facial expression and under varying pose. It is encouraging that the proposed face recognition technique performs consistently superior to (or equally well as) the LEM method and the method based on Eigenfaces (PCA) in all comparison experiments. In general, CEM combination can achieve the best performance level. However, in some cases, CEM intensity outperforms CEM combination: this is when the relative rotation of the faces is too large. On average the recognition rate increases by 10.15% for face recognition under controlled condition, by 36.41% for face recognition under size variation, by 35.84% for face recognition under varying pose, by 8.41% for face recognition under varying lighting and by 29.12% for face recognition under varying facial expression.

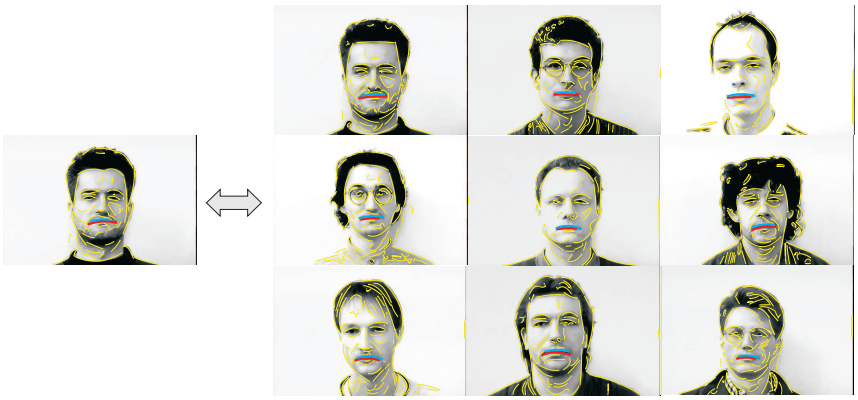


Figure 4.4: Result of facial feature classification using local curve matching. We match the face input of a person with the face model of the same person and with eight other persons. For matching the faces of the same person, we get the parabolas of the mouth with a very low cost, when we match with faces of other persons we mostly get the parabolas of the mouth, but with a higher cost.

We assume that the main reason for the better results of CEM matching are the larger and more stable parabolas, where each parabola describes a significant part of the face. Many parabolas correspond to physically meaningful features, which is illustrated in Figure 4.4 where we match the face input of a person with the face model of the same person and with eight other persons. For matching the faces of the same person, we get the parabolas of the mouth with a very low cost, when we match with faces of other persons we mostly get the parabolas of the mouth, but with a higher cost. Since a parabola has a more distinctive shape than a line segment, matching parabolas is more reliable. Furthermore, each parabola defines a convex region, so that it makes sense to compute the average intensity difference between the convex and concave side of a parabola. The recognition rate improves considerably by the introduction of intensity variations in the distance measure.

Global curve matching

The face recognition rates for global curve matching are given in Table 4.2. A distinction is made between matching using histograms of relative positions, matching using intensity histograms and matching using an optimal linear combination of both. In the results, a match is only correct when the best matched face from a model is the correct person. We test face recognition under controlled condition and size variation, under varying lighting condition, under varying facial expression and under varying pose. As we expect, the matching of histograms with relative positions is more sensitive to varying pose and varying facial ex-

(%)	Hist. of rel. pos.	Intensity hist.	Lin. comb.
Controlled conditions			
GTFD	82.00	98.00	100.00
ATT	77.50	95.00	98.50
YFD	86.66	100.00	100.00
BERN	93.33	96.67	100.00
AR	78.65	96.02	98.44
Pose variation			
BERN Right	70.33	87.00	91.33
BERN Left	70.67	88.33	91.67
BERN Up	72.00	86.67	90.33
BERN Down	69.00	80.33	83.67
Size variation			
AR with size variation	77.83	89.20	94.35
Lighting condition variation			
AR with left light on	73.53	91.89	96.60
AR with right light on	73.43	89.16	95.63
AR with both lights on	75.15	90.63	96.38
Facial expression variation			
AR with smiling expr.	71.13	93.70	97.13
AR with angry expr.	71.59	94.14	97.46
AR with screaming expr.	72.08	91.40	96.65
Average	**76.18**	**91.76**	**95.51**

Table 4.2: The columns in this table show the face recognition rates of our method using histograms of relative positions, matching using intensity histograms and matching using an optimal linear combination of both, respectively. We test face recognition under controlled condition and size variation, under varying lighting condition, under varying facial expression and under varying pose. The matching of histograms with relative positions is more sensitive to varying pose and varying facial expressions, while the matching of intensity histograms is more sensitive to varying lighting conditions. The average face recognition for the linearly combined cost is 95.51%.

pressions, while the matching of intensity histograms is more sensitive to varying lighting conditions. The average face recognition for the linearly combined cost is 95.51%. When compared to the results described for the local curve matching technique in Table 4.1, the average face recognition rate increases with 2.21%. Furthermore, we gain the advantage of facial feature detection, which involves facial action recognition, such as recognition of facial expressions, speaker detection and head movement detection.

4.2.4 Facial feature classification

As mentioned in Chapter 1, we consider a top-down approach. Firstly faces are recognized. Then, individual facial features are classified. Facial features such as the eyebrows, the eyes, the nose and the lips, are detected in frontal face images by matching face CEMs with polynomial curve models. This work was published in [Deboeverie et al., 2011].

Firstly, polynomial curves from the facial features of interest have been modelled by a training process on a database, consisting of 100 faces, considering small variations in pose, variations in lighting conditions and variations in facial expressions. The polynomial curve model of a facial feature consists of two intensity histograms from the inner and outer side of the polynomial curve and a log-polar histogram describing the relative positions in the face CEM.

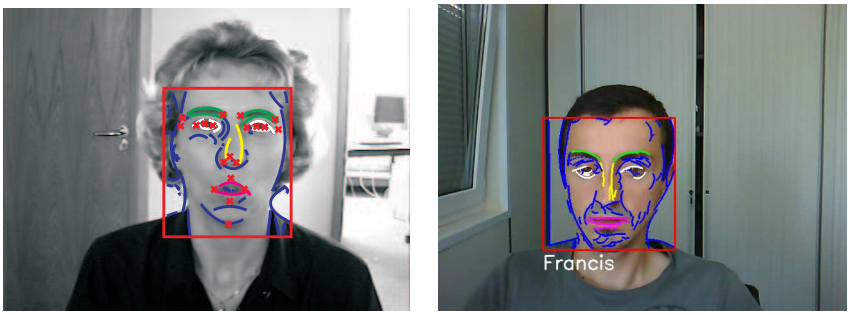
Secondly, the polynomial curve models are matched with the input face CEM. The histograms of intensities and the histograms of relative positions are compared. E.g. for the upper polynomial curve of a left eyebrow we expect a transition in intensity from bright to dark and with a relative position in the left upper corner in the face CEM. In this way, we classify the polynomial curves from the left and right eyebrow, the left and right eye, the left and right side of the nasal bone and the upper and the lower lip.

We evaluate this facial feature classification on the test databases by verifying whether or not the polynomial curve models classify the correct polynomial curves in the face CEMs. We classify the polynomial curves from the left and the right eyebrow, the left and the right eye, the left and the right side of the nasal bone and the upper and the lower lip, as shown in Figures 4.5 (a) and (b). The results of facial feature detection, as presented in Table 4.3, show that the developed system can detect on average the individual facial features successfully in 91.92% cases, when applied to the test databases. We compare our results with the work of Cristinacce et. al. [Cristinacce and Cootes, 2008], in which they use the Viola and Jones face detector [Viola and Jones, 2001] to initialize a set of facial feature points. The Viola and Jones face detector finds 95% of facial feature points within 20% of the inter-ocular separation on the BIOID Database. Our average facial feature detection rate on the BioID Database is 93.30%, which is comparable with the facial feature detection rate by applying the Viola and Jones face detector.

The accuracy in position of the facial features with the polynomial curves is determined by the accuracy in position of the edges delivered by the Canny edge detector and the fitting cost allowed during contour segmentation. We compute this accuracy on the BioID Database, by comparing the available ground truth marker points with the locations of the facial features. In Figure 4.5 (a), the ground truth marker points are indicated by red coloured crosses. We define the accuracy in position of the facial features by the distance between the points on the polynomial curves closest to the ground truth markers and the ground truth markers them-

(%)	YFD	BERN	AR	GTFD	ATT	BioID
Left eyebrow	100.00	90.00	87.91	96.00	92.50	93.12
Right eyebrow	100.00	80.00	90.47	94.00	90.00	94.65
Left eye	93.33	90.00	86.20	86.00	85.00	91.39
Right eye	93.33	93.33	87.91	88.00	87.50	90.94
Left nose	100.00	90.00	88.03	96.00	92.50	94.10
Right nose	100.00	96.67	87.18	90.00	95.00	92.94
Upper lip	93.33	90.00	89.20	86.00	97.50	95.51
Lower lip	100.00	96.67	90.60	86.00	92.50	93.77
Average	**97.50**	**90.46**	**88.44**	**90.25**	**91.56**	**93.30**

Table 4.3: An overview of the facial feature detection rates of the left and the right eyebrow, the left and the right eye, the left and the right side of the nasal bone and the upper and lower lip. The system can detect on average the individual facial features successfully in 91.92% cases.



(a) (b)

Figure 4.5: (a): The polynomial curves of the eyebrows, the eyes, the nose and the lips detected in a face image of the BioID face Database. The ground truth marker points are indicated by red coloured crosses. (b): Facial feature classification using global curve matching in a face image from a webcam video sequence.

selves. When using the Euclidean mean distance as the basic error measurement, the mean error is 6.53 pixels with a standard deviation of 1.15 pixels, which is 2.18% in terms of the size of the face. On the same database and at the same image resolution, we compare our results to the work of Ding et. al. [Ding and Martinez, 2010], in which they achieve facial feature detection with SubAdaBoost by learning the textural information and the context of the facial feature to be detected. Here, the mean error is 9.0 pixels with a standard deviation of 1.3 pixels, which is 2.8% in terms of the size of the face.

4.3 Best view selection

Nowadays, an important problem in multi-camera systems is how to select the camera with the best frontal view of a person. Therefore, we present a minimum score based criterion for best view selection, based on face recognition with geometric features. In this approach, faces are represented with CEMs, as described in Section 2.5. Face recognition is performed by matching face CEMs driven by histograms of intensities and histograms of relative positions, as described in Section 3.3.2. The resulting face recognition scores are employed as quality-of-view measures. The face recognition scores reach a minimum for frontal face views. They indicate whether or not persons are seen by cameras in frontal view. Experiments show that the method is robust and efficient when selecting the best view in a multi-camera system. Furthermore, our method outperforms view selection based on face detection only. This work was published in [Deboeverie et al., 2012].

In automated video conferencing, persons are observed by several cameras. Algorithms for behaviour analysis deliver information about who is attracting attention, for instance someone who is making hand gestures. From this information, a virtual director can decide which person has to be visualized. The question remaining is: which camera has the best frontal view of the person of interest? In this section, we show that for each person-camera combination a face recognition score, obtained with geometric features, is a good quality-of-view measure. This information can be used for visualization or it can be fed to a close-up face analysis system, which uses it to only selectively process some areas of interest within some video streams, for example to detect a speaker, to determine eye gaze and to refine the face orientation analysis.

4.3.1 Related work on best view selection

Viewpoint selection has been studied in the fields of computer graphics and robot navigation [Vázquez et al., 2003, Roberts and Marshall, 1998]. More directly related to this work is [Feris et al., 2007], where a single camera collects key frames of people in surveillance video based on face detections. View selection for observability is treated in [Daniyal et al., 2010, Jiang et al., 2008, Kelly et al., 2009, Li and Bhanu, 2009, Tessens et al., 2008]. The authors in [Daniyal et al., 2010] assign a score to the content of each view by measuring the activity level, the number of objects, events, etc. The size of the bounding box of an object is used as a quality of view measure in [Jiang et al., 2008], where dynamic programming is used to optimize the selection over time. The object size and centrality in the camera image are considered in [Kelly et al., 2009], complemented by a face detection measure in [Li and Bhanu, 2009]. In [Tessens et al., 2008], view point selection was based on the position and motion of observed persons and on the visible face area.

In this work, we propose to use a face recognition reliability measure with

geometric features as a criterion for view selection. This is new because face recognition has not been successfully exploited yet for view selection.

4.3.2 Method of best view selection

The goal of best view selection is to determine a view that best shows a person from many available views in a vision network. The system we consider consists of multiple cameras in a meeting room. The cameras are denoted by C_i for $i = 1, \ldots, N$, with N the total number of sensors. The image captured by the i-th camera at a certain time instant t is denoted by $I_i(t)$. The different persons are denoted by P_j for $j = 1, \ldots, L$, with L the total number of persons in the scene. The key or principal camera is the camera with the view that contributes most to the desired observation of the person at a certain time instant, i.e., that captures a frontal view of a person in the scene. Persons observed with multiple cameras will almost always be seen by one of the cameras in frontal or nearly frontal view.

The algorithms for CEM extraction and face recognition with face CEMs are run on the images $I_i(t)$ captured at a certain time instant t for each camera C_i. At each time instant t, the face recognition algorithm returns the following values: $z_i(t)$ and $Q_i^l(t)$, for $l = 1, \ldots, z_i(t)$. $z_i(t)$ is the number of faces recognized in the frame $I_i(t)$. $Q_i^l(t)$ is a measure of the quality of the l-th recognized face. In our implementation, $Q_i^l(t)$ is the matching score of the face recognition with face CEMs. The higher this value, the less certain the recognition.

In this method for best view determination the face recognition score $Q_i^l(t)$ is used to select the principal view per person. To deal with spurious face recognitions and to obtain smoothness over time, the selection of the key camera at time instant t not only depends on the current face recognition output, but also on the previous observations. For each camera C_i, the temporally filtered face recognition score $R_i^l(t)$ is an exponentially weighted moving average of the current observation and the previous temporally filtered face recognition score $R_i^l(t-1)$, with $R_i^l(0) = 0$:

$$R_i^l(t) = \alpha Q_i^l(t) + (1-\alpha) R_i^l(t-1), \forall t \geq 1 \tag{4.1}$$

where α is a constant between zero and one that determines the importance of previous observations. Then, the best view camera for a person P_l at time instant $t \geq 1$ is

$$S^l(t) = \underset{C_i}{\operatorname{argmin}}\, R_i^l(t) \tag{4.2}$$

4.3.3 Evaluation of best view selection

In this paragraph, we assess the performance of the proposed best view selection method for observability.

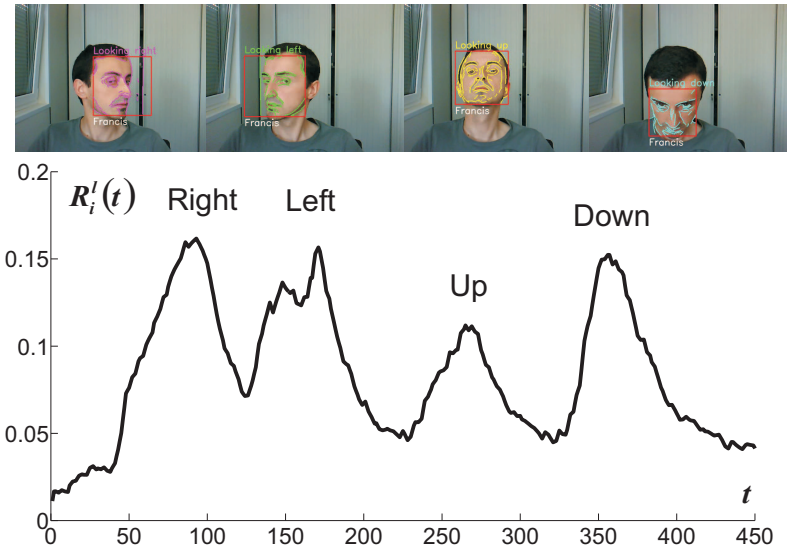


Figure 4.6: This graph plots the view measure scores $R_i^l(t)$ for a head which is rotated right, left, up and down, respectively. The face recognition keeps working when the face is not parallel to the camera. However, $R_i^l(t)$ clearly increases in that case. Consequently, the more frontal a face view, the smaller $R_i^l(t)$.

For a rotating head, we examine the face view measure scores $R_i^l(t)$ in Eq. (4.1). The images in Figure 4.6 show a face which is rotated right, left, up and down, respectively. $R_i^l(t)$ is plotted in the graph in Figure 4.6. The face recognition keeps working when the face is not parallel to the camera. However, $R_i^l(t)$ clearly increases in that case. Consequently, the more frontal a face view, the smaller $R_i^l(t)$.

Test case

Experimental video data for testing the method on, was recorded with a camera setup as shown in Figure 4.7. This test case considers a meeting with $L = 8$ persons, recorded with $N = 4$ HD-cameras in the corners. In this meeting there was a lot of interaction between the persons, accompanied by explicit head gestures.

The temporal filtering parameters of the best view selection is set to $\alpha = 0.05$. This parameter has been manually tuned on a small number of frames. α has been set to a small value, such that previous observations are weighted heavily. When the value of α is increased, the best view will be switched more frequently. To evaluate the quality of the best view selected by our method, we use sequences labelled by human observers as a benchmark. The total number of labelled frames is 5000.

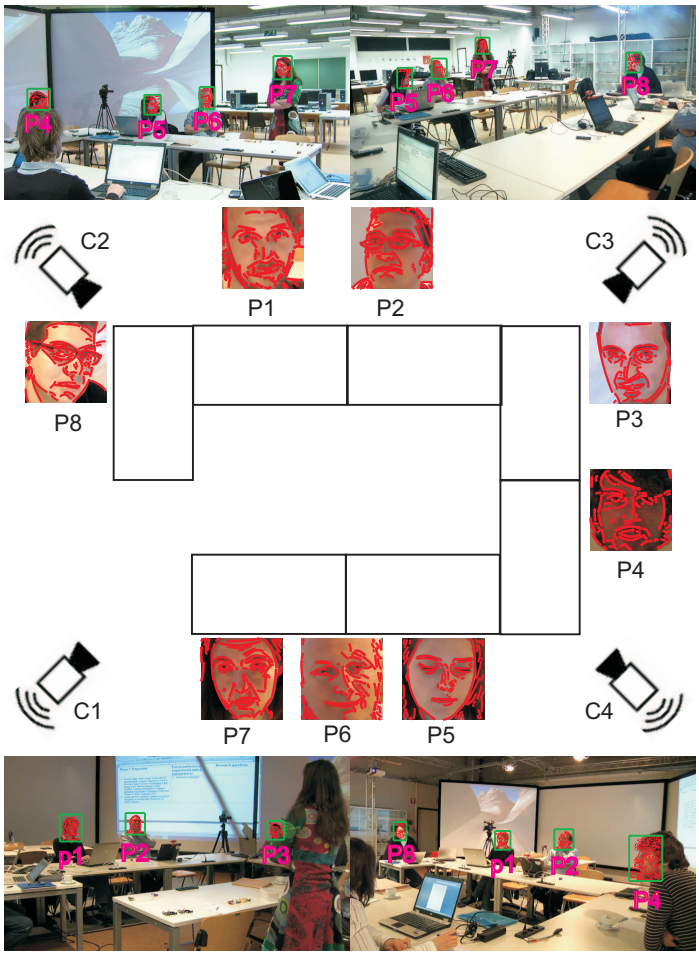


Figure 4.7: Test scenario: a meeting with 8 persons, recorded with 4 HD-cameras in the corners.

We compare our algorithm with an algorithm which uses face *detection* rather than recognition as a measure for best view selection [Li and Bhanu, 2009]. The face detection detects the frontal faces with the object detector that was initially proposed in [Viola and Jones, 2001]. The visible face area is the measure for best view selection.

Some typical examples of correct and incorrect camera selections C_i of the best view per person P_i are shown in the graphs in Figure 4.8. The face recognition scores in the cameras $C1, \ldots, C_4$ are indicated by distinct grey values. The dotted, dashed and dash-dotted vertical lines indicate camera switches based on

Scenario as in Figure 4.7	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
Face detection	73	72	75	61	51	55	72	75
Face detection + recognition	81	87	84	90	64	69	86	80

Table 4.4: The percentage of frames in which the view selected as best view per person by the methods based on face detection cues and based on face detection and recognition cues were labelled as a best one by a human observer. The best view selection based on face recognition cues achieves a better hit rate for each person in the scene than best view selection based on face detection cues.

face recognition, face detection and ground truth, respectively. Ground truth is obtained by best view labelling per person by a human observer. The graph in Figure 4.8 (a) shows a correct camera switch when P_1 turns his head from C_1 to C_4 and back again. The graph in Figure 4.8 (b) shows a correct camera switch when P_7 is turning his head from C_2 to C_3 and back again. The graph in Figure 4.8 (c) shows that the recognition of P_8 in C_3 and C_4 is nearly equal. In the beginning, there are a lot of undesirable camera switches for the observer. These are avoided by allowing only one camera switch over a short time interval.

Table 4.4 indicates the percentage of frames in which the view selected as best view per person were labelled as best by a human observer. Comparing the results from the method based on face detection cues and our method based on face detection and recognition cues, we conclude that our method provides a powerful means to boost the hit rate. We can also observe in Table 4.4 that the best view selection based on face recognition cues achieves a good hit rate for each person in the scene, or in other words, that it very often selects the view which also a human observer judges as providing a good observation of the persons in a scene.

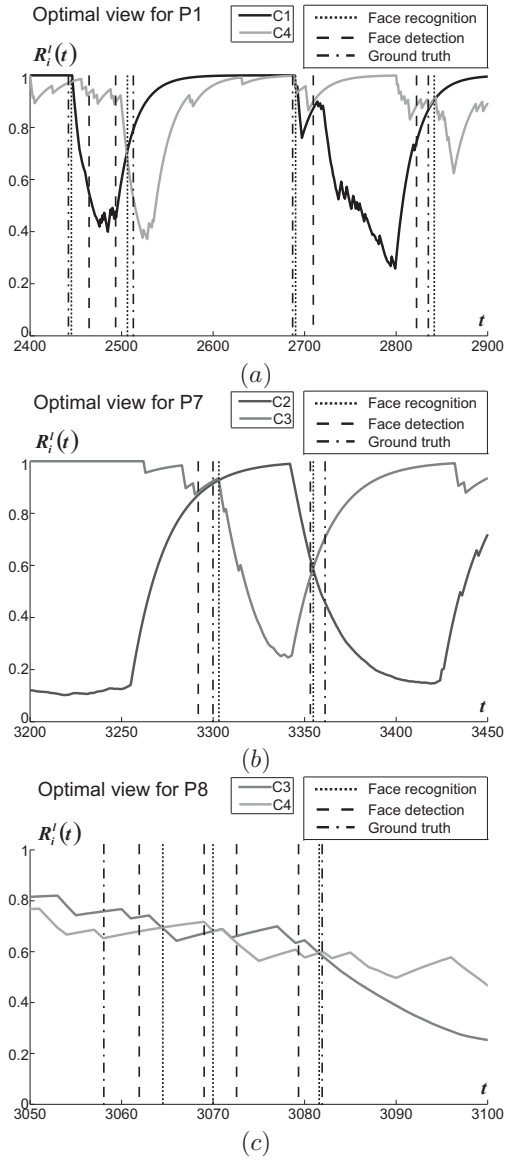


Figure 4.8: Some examples of correct and incorrect selections of the best view per person. The face recognition scores in the cameras $C1, \ldots, C_4$ are indicated by distinct grey values. The dotted, dashed and dash-dotted vertical lines indicate camera switches based on face recognition, face detection and ground truth, respectively.

(a) (b)

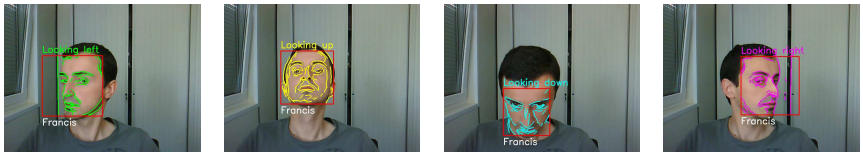
Figure 4.9: Result of entering/leaving detection. In consecutive frames, we successfully track the faces of two persons (green bounding boxes). The bounding boxes are getting larger, such that the behaviour of the persons is classified as entering the room. Sometimes the face detection algorithm of Viola and Jones [Viola and Jones, 2001] detects non-faces (false positives). Then, the tracking does not find corresponces in consecutive frames for these non-faces, such that they are classified as outliers (red bounding boxes).

4.4 Behaviour analysis applications

In this section, we present a few visual results of behaviour analysis applications, such as detection of people entering or leaving a room, head movement detection and speaker detection.

4.4.1 Entering/leaving detection

Entering/leaving detection is firstly performed by face tracking using polynomial curve matching as described in Section 3.3.1. Next, a bounding box which is getting larger or smaller defines if a person is entering or leaving, respectively. A visual example of entering detection is found in Figure 4.9. In consecutive frames, we successfully track (find correspondences) for the faces of two persons (green bounding boxes). The bounding boxes are getting larger, such that the behaviour of the persons is classified as entering the room. Sometimes the face detection algorithm of Viola and Jones [Viola and Jones, 2001] detects non-faces (false positives). Then, the tracking does not find corresponces in the consecutive frames for these non-faces, such that they are classified as outliers (red bounding boxes).



(a) (b) (c) (d)

Figure 4.10: Result of head movement detection. Images (a), (b), (c) and (d) show a head from a webcam video sequence, which is moving left, up, down, and right, respectively.

4.4.2 Head movement detection

Head pose estimation is the process of inferring the orientation of a human head from digital imagery [Murphy-Chutorian and Trivedi, 2009]. In this work, head pose estimation or head movement detection is firstly performed by face tracking using polynomial curve matching as described in Section 3.3.1. From the motion vectors of the curve segment pairs, an average motion vector defines in which direction a person's head is moving, as described in Section 3.3.1.5. Then, if the length of the head motion vector is larger than a predefined threshold, the head is classified as moving, otherwise the head is classified as not moving. The direction of the head motion vector classifies if the head is moving up, down, right or left. For a face size of 200x200 pixels and a frame rate of 25 frames per second, an actual value of the threshold for head movement classification is 5 pixels. Figures 4.10 (a), (b), (c) and (d) show visual results for head movement detection of a face, which is moving left, right, up and down, respectively.

We evaluate head movement detection on an video sequence of a reporter during a news interview. We manually annotated 130 head movements, of which 45 are moving up, down, left or right and 85 are not moving. The graph in Figure 4.11 (a) diplays the lengths of the average head motion vectors over 1200 frames. When listening to questions, the reporter is sometimes nodding to indicate that she has understood the question. The nodding is also clearly visible in the directions of the average head motion vectors, as shown in Figure 4.11 (b). An example of the head of the reporter that is moving down and up during nodding is shown in Figures 4.12 (a) and (b), respectively. When answering questions, the reporter is making small head gestures to reinforce her response.

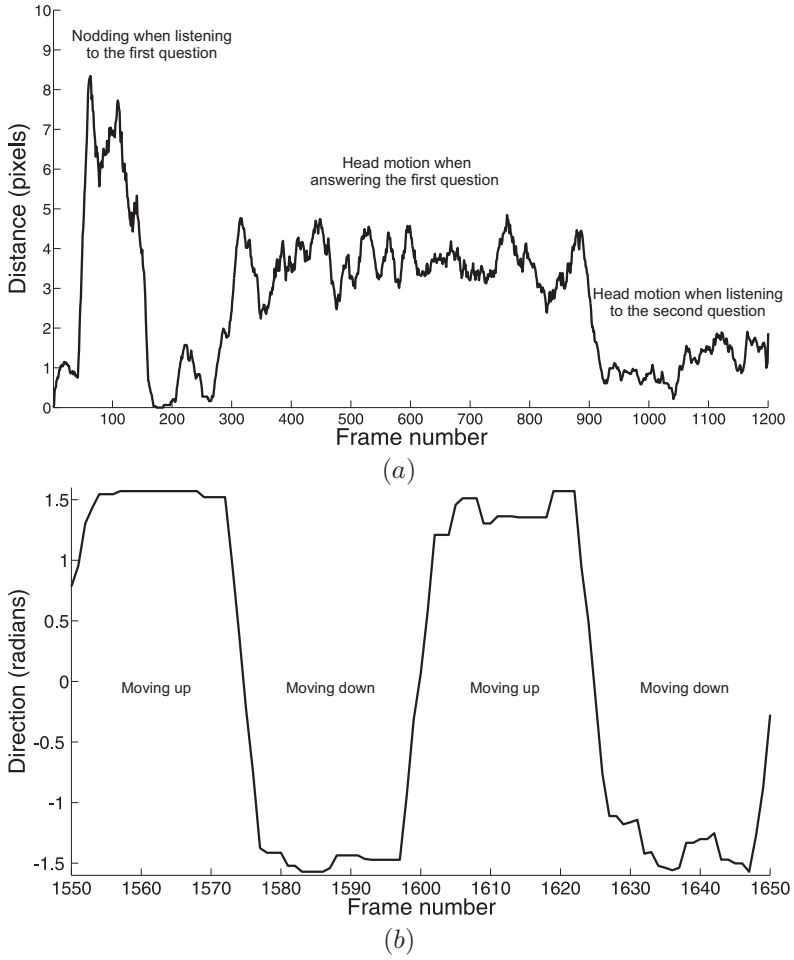


Figure 4.11: (a) and (b): Lengths and directions of the head motion vectors during an interview. When listening to questions, the reporter is sometimes nodding to indicate that she has understood the question. When answering questions, the reporter is making small head gestures to reinforce her response.

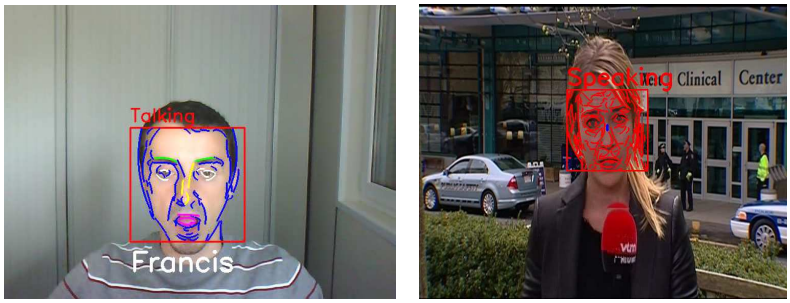
(a) (b)

Figure 4.12: (a) and (b): An example of the head of a reporter that is moving down and up during nodding, respectively.

Head movement detection	TP	FN	TN	FP
Ours	39	6	75	10
[Cootes et al., 2002]	37	8	76	9

Table 4.5: The results of TP, FN, TN and FP for head movement detection performed by our method and head pose estimation based on the Active Appearance Model (AAM) [Cootes et al., 2002]. Our method classifies head motion correctly in 87,69% cases, which is comparable with the correct classification percentage of 86,92% of the method based on AAM.

Table 4.5 shows the results of True Positives (TP), False Negatives (FN), True Negatives (TN) and False Positives (FP) for head movement detection performed by our method and for head pose estimation based on an Active Appearance Model (AAM) [Cootes et al., 2002], where a TP is a correct classification of a head that is moving up, down, left or right, a FN is an incorrect classification of a head that is moving up, down, left or right, a TN is correct classification of a head that is not moving, and a FP is an incorrect classification of head that is not moving. The AAM learns the primary modes of variation in facial shape and texture from a 2D perspective. Then, an estimate of head pose can be obtained by mapping the appearance parameters to a pose estimate. Our method classifies head motion correctly in 87,69% cases, which is comparable with the correct classification percentage of 86,92% of the method based on AAM.



(a) (b)

Figure 4.13: (a) and (b): Two examples of automated speaker detection.

4.4.3 Speaker detection

Visual Voice Activity Detection (VAD) is the detection of speech frow a video sequence by means of visual cues [Aubrey et al., 2010]. In this work, visual VAD or speaker detection is firstly performed by detection of the polynomial curves approximating the lips, as described in Section 4.2. The next step is motion registration of the polynomial curves describing the lips, as described in Section 3.3.1.5. Then, if the absolute difference between the vertical length of the motion vectors of the upper lip and the lower lip is larger than a predefined threshold, the person is classified as speaking. For a face size of 200x200 pixels and a frame rate of 25 frames per second, an actual value of the threshold is 2 pixels. A visual example of automated detection of a speaking person is shown in Figure 4.13 (a).

We evaluate speaker detection on the same video sequence as in the previous section of head movement detection. We manually annotated 150 speaking actions, of which 54 are speaking and 96 are not speaking. An example of the speaking reporter is shown in Figure 4.13(b). The reporter shows increased lip motion when answering questions as displayed in the graph in Figure 4.14. Table 4.6 shows the results of True Positives (TP), False Negatives (FN), True Negatives (TN) and False Positives (FP) for speaker detection performed by our method and for speaker detection based on an Active Appearance Model (AAM) [Aubrey et al., 2007]. The method based on AAM uses appearance parameters of a speaker's lips. Then, a Hidden Markov Model (HMM) dynamically models the change in appearance over time. Our method classifies speaker detection correctly in 74,00% cases, which is comparable with the correct classifaction percentage of 73,33% of the method based on AAM. In general, the results of head movement detection are better than those of speaker detection. This is because head movement detection considers a large movement of the entire face, while speaker detection considers only a small movement of a small part of the face.

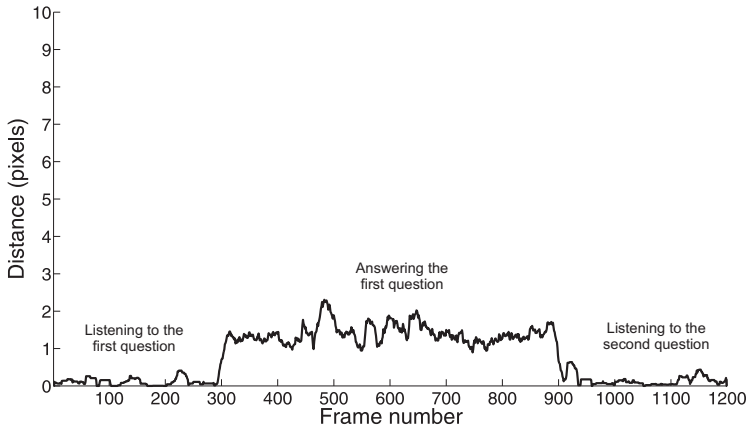


Figure 4.14: The absolute difference between the vertical length of the motion vectors of the upper lip and the lower lip while speaking during an inteview.

Speaker detection	TP	FN	TN	FP
Ours	42	12	69	27
[Aubrey et al., 2007]	40	14	70	26

Table 4.6: The results of TP, FN, TN and FP for speaker detection. Our method classifies speaker detection correctly in 74,00% cases, which is comparable with the correct classification percentage of 73,33% of the method based on AAM.

4.5 Conclusion

In this chapter, we evaluated the matching techniques for polynomial curves, as proposed in Chapter 3, by face recognition and tracking. We considered security applications, such as people identification and best view selection, and behaviour analysis applications, such as entering/leaving detection, head movement detection and speaker detection. We evaluated the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compared the proposed methods with several techniques of the state of the art. The face analysis results are comparable with or better than existing methods. The advantages of our methods are simplicity, real-time properties and many face analysis tasks are handled by the same approach.

This research was published in four papers [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2011, Deboeverie et al., 2012] in the proceedings of international conferences.

5
Tracking of other objects

5.1 Introduction

Object tracking is the problem of identifying and following image elements moving across a video sequence automatically. It has attracted much attention due to its many applications in computer vision, including surveillance, perceptual user interfaces, augmented reality, smart rooms, driver assistance, medical imaging and object-based video coding. Since many applications have real-time requirements, very low computational complexity is a highly desirable property. However, also accuracy is very important. Thus, it is of interest to develop an object tracking framework that can address all of these diverse requirements.

In this chapter, we propose a method for moving object detection and tracking. Firstly, we represent scene observations with CEMs, as described in Section 2.5. Secondly, we obtain motion vectors for these polynomial curves in consecutive frames by a matching technique based on distance and intensity, as described in Section 3.3.1. Then, moving objects are detected by an original method that clusters comparable motion vectors, as described in Section 3.3.1.5. The result is a robust detection and tracking method, which can cope with small changes in viewpoint on the moving object. In this chapter, we consider tracking of other objects than faces, such as vehicles, heart walls and water currents.

The work in this chapter was published in [Deboeverie et al., 2009a,Deboeverie et al., 2009b].

This chapter is structured as follows: tracking of vehicles, heart walls and

water currents is considered in Sections 5.2, 5.3 and 5.4, respectively.

5.2 Vehicle tracking

Applications such as traffic surveillance require a real-time and accurate method for object tracking. We propose detection and tracking of vehicles with CEMs. This work was published in [Deboeverie et al., 2009b]. We use motion vectors to describe the motion of polynomial curves, which are matched one-to-one by a technique that considers both geometric distance and intensity profile similarity, as described in Section 3.3.1. Moving vehicles are detected and tracked by a cluster algorithm for sets of comparable curve motion vectors, as described in Section 3.3.1.5. We will show by experimental results that our method is stable for small angle and scale changes, which is advantageous in the tracking of vehicles in bends.

Related work on vehicle tracking

An overview of general object detection is described in 3.2. Regarding vehicles, a large number of studies have been devoted to vehicle detection and tracking. Jelaca et al. [Jelaca et al., 2013] represent vehicle appearances using signature vectors composed of Radon transform like projections of the vehicle images and compare them in a combination of 1-D correlations. To deal with appearance changes they include multiple observations in each vehicle appearance model. Rios-Cabrera et al. [Cabrera et al., 2012] use Haar-like features for vehicle matching. These features are often successfully used for object detection [Viola and Jones, 2001], so reusing the same features for matching reduces the computational cost of the matching itself. The most informative Haar features are selected by a supervised Ada-Boost training in several cascades. Binary vehicle fingerprints embedded from those same Haar features are used to match vehicles, as well as to track vehicles in a tracking-by-identification fashion. Shan et al. [Shan et al., 2008] has proposed a measurement vector and an unsupervised approach to learn edge measures for matching vehicle edge maps. The edge maps are compared after spatial alignment. Wang et al. [Wang and Yagi, 2008] extended the standard mean-shift tracking algorithm to an adaptive tracker by selecting reliable features from colour and shape-texture cues according to their descriptive ability. Using the shape for image understanding is a growing topic in computer vision and multimedia processing and finding good shape descriptors and matching measures is the central issue in these applications. Xu et al. [Xu et al., 2009] proposed a shape descriptor of planar contours which represents the deformable potential at each point along a contour. Ferrari et al. [Ferrari et al., 2008] presented a family of scale-invariant local shape features formed by chains of connected roughly straight contour segments, and their use for object class detection.

Xiong et al. [Xiong and Debrunner, 2004] examined real-time vehicle tracking by combining a colour histogram feature with an edge-gradient-based shape feature under a sequential Monte Carlo framework. The difficulties that these methods have to face are mainly viewpoint, illumination changes and appearance changes, which in some applications have to be addressed in real time.

Method of vehicle tracking

In this work, vehicle tracking is initialized by moving object detection. A block diagram with the method overview of moving object detection is shown in Figure 5.1. Firstly, we extract CEMs in two consecutive frames at time instances $t-1$ and t, respectively. The method to extract CEM is thoroughly explained in Section 2.5. Then, two CEMs of the scene are matched using local curve matching as described in Section 3.3.1. As last step, moving vehicles are detected by a cluster algorithm for sets of comparable curve motion vectors, as described in Section 3.3.1.5. The result is a CEM of the vehicle at time instance t. A resulting bounding box which encloses all polynomial curves of the CEM is computed.

Once detected, vehicles are tracked in consecutive frames. A block diagram with the method overview of object tracking is shown in Figure 5.2. In a frame at time instance t, we extract a CEM in the spatial neighbourhoud of the vehicle at time instance $t-1$. Then, the CEM of the vehicle at $t-1$ is matched with the local CEM at t. From the curve correspondences, the CEM of the vehicle at t is extracted by an update of curves in the CEM of the vehicle at $t-1$: curves at $t-1$ without correspondences are removed from the cluster, while curves at t with correspondences are introduced in the cluster.

Evaluation of vehicle tracking

The proposed methods for vehicle detection and tracking are evaluated on different video sequences. We consider real traffic surveillance videos in tunnels (Seq. 1 with a video resolution of 720x576 pixels), as well as moving vehicles monitored in a parking area (Seq. 2 with a video resolution of 1920x1080 pixels). An example of a vehicle passing of front of the camera in a tunnel in Seq. 1 is given in Figure 5.3 (a). As ground truth, we manually annotated 70 and 20 passing vehicles in Seq. 1 and Seq. 2, respectively.

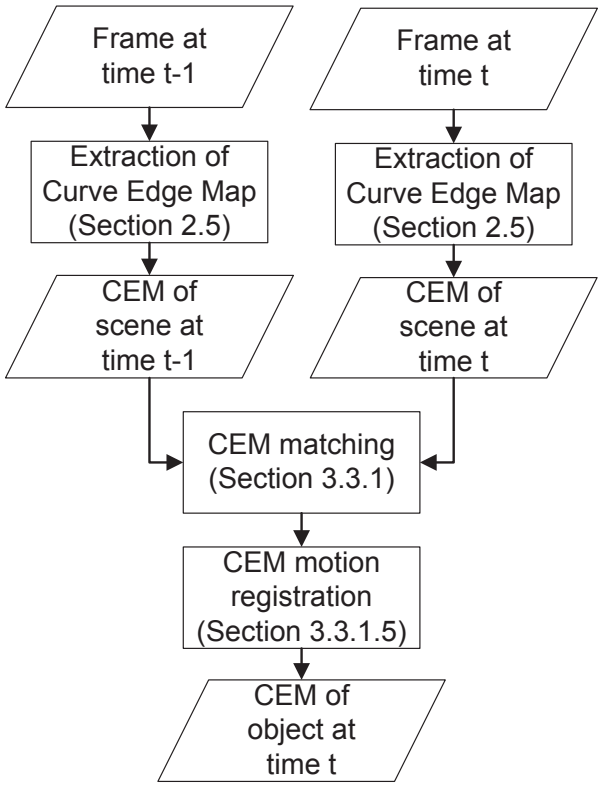


Figure 5.1: This block diagram represents the method overview of moving object detection. Firstly, we extract CEMs in two consecutive frames at time instances $t-1$ and t, respectively. Then, two CEMs of the scene are matched using local curve matching. As last step, moving objects are detected by a cluster algorithm for sets of comparable curve motion vectors. The result is a CEM of the object at time instance t.

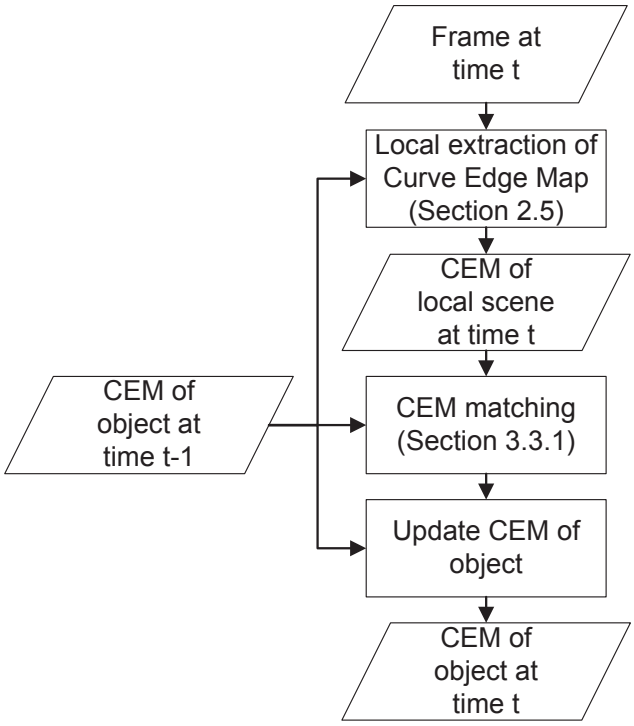
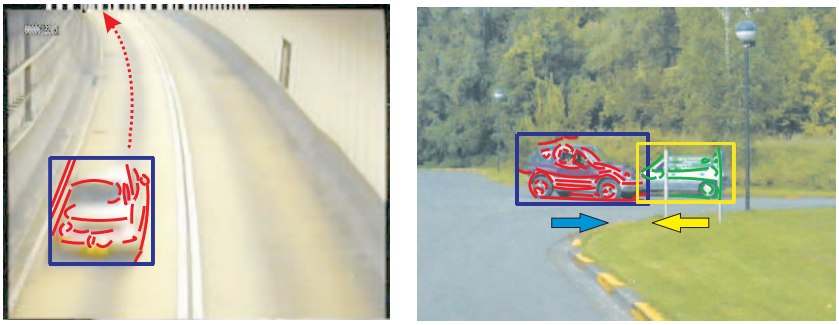


Figure 5.2: This block diagram represents the method overview of object tracking. In a frame at time instance t, we extract a CEM in the spatial neighbourhoud of the object at time instance $t-1$. Then, the CEM of the object at $t-1$ is matched with the local CEM at t. From the curve correspondences, the CEM of the object at t is extracted by an update of curves in the CEM of the object at $t-1$: curves at $t-1$ without correspondences are removed from the cluster, while curves at t with correspondences are introduced in the cluster.



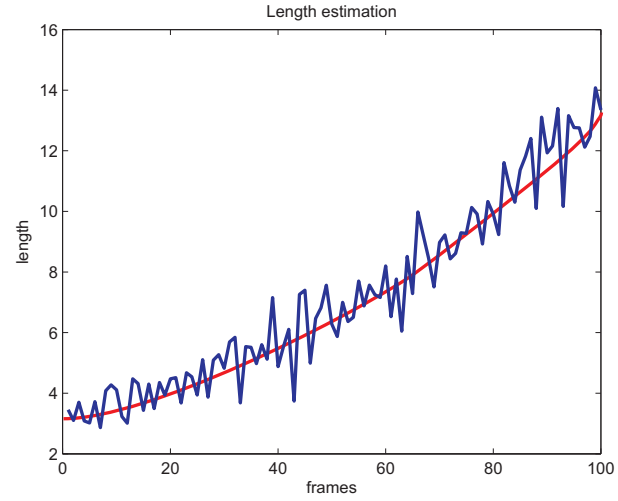
(a) (b)

Figure 5.3: (a): An application example of tunnel surveillance: a vehicle passing by in the camera view in a tunnel. (b): An example of occluding polynomial curve clusters.

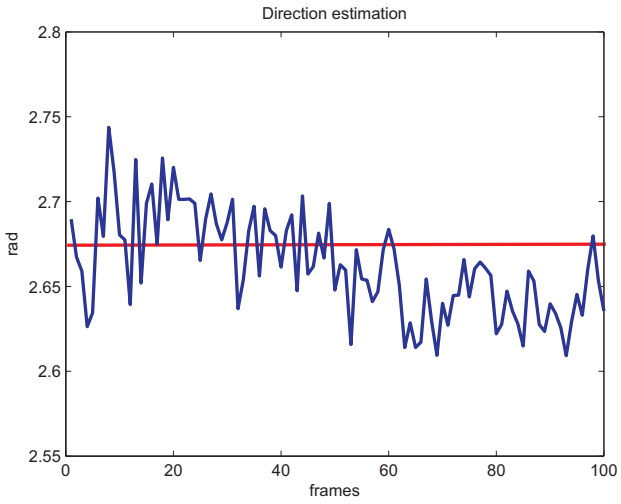
	Vehicle detection (%)		Vehicle tracking (%)	
	Seq 1	Seq 2	Seq 1	Seq 2
Based on CEM features	84,29	95,00	78,57	85,00
Based on Haar-like features	72,86	85,00	67,14	70,00

Table 5.1: Results of vehicle detection and tracking on two video sequences (Seq. 1 and Seq. 2) of our method based on CEM features and of a method based on Haar-like features, respectively. We can clearly notice that the method based on CEM features outperforms the method based on Haar-like features. The reason for this is that the method based on CEM features can better handle changes in viewpoint, illumination changes and appearance changes.

We count a correct vehicle detection if 1) the vehicle is detected correctly in the first 25 frames (within 1 second) since the vehicle was entirely visible, 2) the enclosing bounding bounding box at the entrance zone has at least 80% overlap with the bounding box from the ground truth. Beside, we count a correct vehicle track if 1) the vehicle is detected correctly at the entrance zone, 2) the vehicle is followed correctly from the entrance zone to the exit zone, 3) the enclosing bounding bounding box at the exit zone has at least 80% overlap with the bounding box from the ground truth. We make a comparison with vehicle detection and tracking based on Haar-like features. Table 5.1 presents the results for vehicle detection and tracking. We can clearly notice that the method based on CEM features outperforms the method based on Haar-like features. The reason for this is that the method based on CEM features can better handle changes in viewpoint, illumination changes and appearance changes.

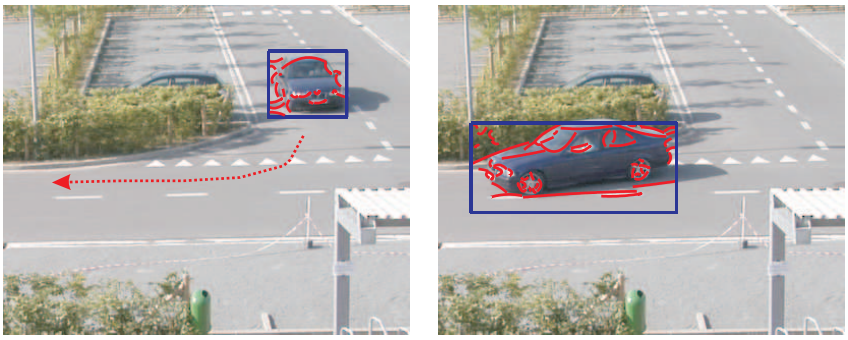


(a)



(b)

Figure 5.4: Graphs plotting the length and direction estimates of the vehicle motion vectors for the trajectory of a vehicle. (a): Length estimates in pixels of the vehicle motion vectors for the trajectory of a vehicle (blue line) against the ground truth (red line). The RMSE for the length estimation is 0.4 pixels. (b): Direction estimates in radians of the vehicle motion vectors for the trajectory of a vehicle (blue line) against the ground truth (red line). The RMSE for the direction estimation is 0.07 radians.



(a) (b)

Figure 5.5: Tracking polynomial curve clusters in which the polynomial curve appearance significantly changes throughout the sequence.

The accuracy of the algorithm for clustering computation was used and evaluated on the example application in which a sequence shows a car driving in one lane. Our method is evaluated by comparison to ground truth data, which is manually created by indicating the position of the car throughout the image sequence. To assess the accuracy of the vehicle motion vectors for the trajectory of a vehicle, we compute the RMSE of the length and direction estimates against the ground truth. The graph in Figure 5.4 (a) plots the length estimates in pixels of the vehicle motion vectors for the trajectory of a vehicle (blue line) against the ground truth (red line). The RMSE for the length estimation is 0.4 pixels. The graph in Figure 5.4(b) plots the direction estimates in radians of the vehicle motion vectors for the trajectory of a vehicle (blue line) against the ground truth (red line). The RMSE for the direction estimation is 0.07 radians.

The tracking algorithm for clusters of polynomial curves also proves the work in situations where the visual appearance of the objects is radically changing throughout the video sequence. An example is given in Figures 5.5 (a) and (b), where the vehicle takes a turn, and our method succeeds in tracking the changing polynomial curves.

Our method can be still improved, e.g. one of the problems not completely solved at the moment is occlusion. When part of the vehicle gets obscured by other objects, we cannot keep track of the polynomial curve clusters. An example is shown in Figure 5.3 (b). In our experience throughout experiments, a minimal fraction of the object surface must be visible for our method to work. A possible solution could be offered by a linear prediction of the object's position and size using previous parameters of the affine transformation, computed when a sufficient part of the object is still visible.

5.3 Heart wall tracking

In vivo imaging introduces challenges in automated analysis due to high levels of noise and the variability in appearance of the specimen over time. Breathing of the living animal causes additional complications. Automated tracking of blood cells for intravital microscopy is necessary to increase the efficiency of inflammation research by significantly reducing the number of hours required to analyze data. Automated analysis can also provide more accurate displacement measurements and velocity calculations by removing investigator bias. Currently, the automated analysis of video data of the pumping heart and the link between fluid flow and morphological heart diseases in developing embryonic vertebrate hearts is a hot topic. Specifically, extracting meaningful velocity distribution is a difficult research challenge, which requires the analysis of hundreds of blood cells.

In the human embryo, the heart begins to pump blood about three weeks post conception. At this stage the heart has neither chambers nor valves but only consists of a contractile tube. Little is still known about the driving forces behind cardiac development. One hypothesis states that the morphogenesis of the heart is guided by blood flow patterns. To validate this hypothesis, researchers study optically transparent zebrafish (*Danio rerio*) embryos, which are morphologically comparable to human embryonic hearts at this early stage. Since they are optically transparent, they are of obvious scientific use to study the beating heart. The tracking of blood cells and heart walls is used to assess the effects of medication for premature fetuses.

Method of heart wall tracking

In this section, we propose a new fast and reliable algorithm to derive the cardiac output by tracking the motion of blood cells and heart walls in image sequences. An important property of our algorithm is that the tracking of the heart walls and blood cells are treated jointly and become linked processes. We propose to model the bent heart walls with polynomial curves, as described in Section 2.5. The polynomial curves are tracked over time using a technique that considers both geometric distance and intensity, as described in Section 3.3.1. As we will show in the results, this leads to an accurate estimation and tracking of the heart walls. Compared to active contours we achieve a reduction of the RMSE with 14% for the position of the heart walls. Subsequently, this estimate is used to define the region of interest for the blood cell intensity correlation. To segment blood cells, we model them as circular regions and find them with the Circle Hough Transform (CHT) on greyscale images for a limited range of radii. The circles are tracked with intensity correlation techniques, from which the velocity in the fluid flow is estimated. To improve the reliability, the region of interest in which correlation takes place for each blood cell depends on the position of the heart walls.

The main contribution in this work is that we show that the blood cells and heart walls can be detected and tracked with relatively simple geometric features. The reliability of the tracking comes in part from the coupling of blood cell tracking and heart wall tracking. In the velocity of the fluid flow and in the position of the heart walls, we can distinguish three phases of the pumping mechanism of the heart, namely the suction, the compression and the relaxation phase. The fluid flow reacts with a delay to the motion of the heart walls. We use this relation to achieve a significantly decrease of false matches with 55,5%. This results in a much improved estimation of the velocity in the fluid flow. This work was published in [Deboeverie et al., 2009a].

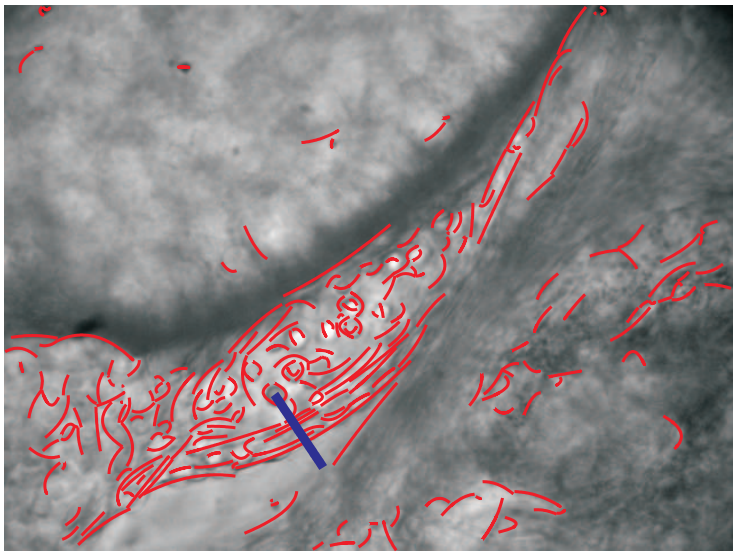
We evaluate our methods with a ground truth, which is manually made by selecting and linking the blood cells and the heart walls. The cell detection and tracking are evaluated over 91 frames, since a cell crosses the heart tube in 91 frames average. However, the heart walls detection and tracking are evaluated over 148 frames, since the position of the heart walls are periodic in 148 frames average.

Related work

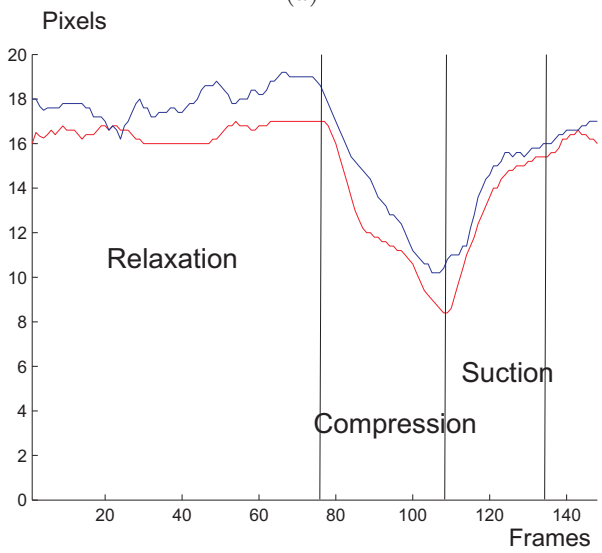
Jeong et al. [Jeong et al., 2006] have proposed a method of computing the velocity and deformation of Red Blood Cells (RBCs) in capillary networks. Recently, Sbalzarini et al. [Sbalzarini and Koutmoutsakos, 2005] developed a feature tracking algorithm for the analysis of particle motion in biological systems. Particles are tracked by minimizing the association cost computed based on distance and trajectory. Eden et al. [Eden et al., 2005] proposed a method of computing flow characteristics of circulating particles by automatically tracking circulating leukocytes in in vivo in larger vessels than capillaries. The cells are detected based on color and temporal features using neural networks. Interestingly, none of these papers treated both the heart wall motion and the blood flow velocity to derive the cardiac output.

Measurement of the position of the heart wall

We measure the positions of the heart wall over time on a manually defined line segment, as shown in Figure 5.6(a). The position is initialized and updated by the intersection point of the outer polynomial curve of the heart wall with the line segment. In a new frame, the motion vectors of the polynomial curves which intersect the line segment, are projected on this line. The current position of the heart wall is the previous position added with an average of the lengths of the projected motion vectors. The resulting curve is shown in Figure 5.6(b). We distinguish the three phases in the positions of the heart wall: the suction, the relaxation and the compression phase.



(a)



(b)

Figure 5.6: (a): The heart wall modelled with polynomial curves, indicated in red colour. We measure the positions of the heart wall over time on the blue line segment. (b): The blue and the red curve correspond to the position estimates of the heart wall and the ground truth, respectively. We distinguish the three phases in the positions of the heart wall: the suction, the relaxation and the compression phase.

To assess the accuracy, we compute the RMSE of the position of the heart wall obtained with polynomial curves against the ground truth. The RMSE is 1.43 pixels, when computed over 148 frames.

Comparison with active contours

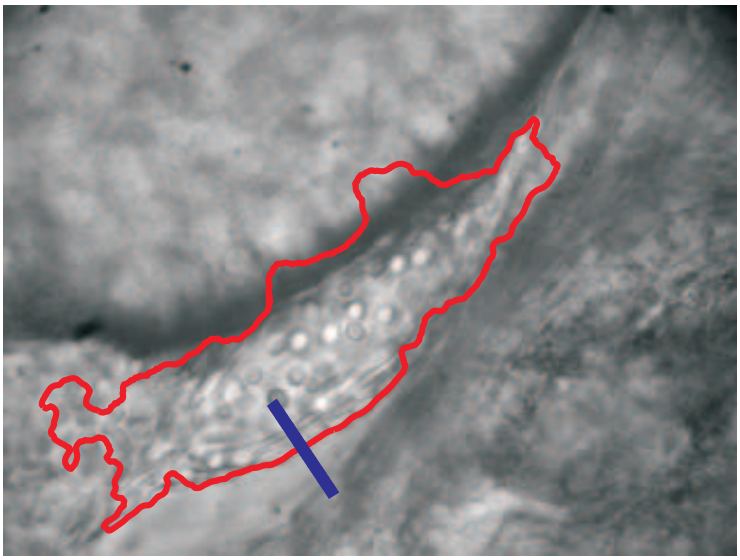
We use the technique from Chunming et al. [Li et al., 2005] for geometric active contours that forces the level set function to be close to a signed distance function, and therefore completely eliminates the need of the costly re-initialization procedure. The variational formulation consists of an internal energy term that penalizes the deviation of the level set function from a signed distance function, and an external energy term that drives the motion of the zero level set toward the desired image features, such as the heart wall. The positions of the heart wall are measured as the positions of the active contour on the line segment, as shown in Figure 5.7(a) and (b).

To assess the accuracy of the positions of the heart wall obtained with active contours, we compute the RMSE of the positions of the active contour against the ground truth. The RMSE is 1.63 pixels, when computed over 148 frames. This is 14% higher than for polynomial curves. As for tracking, the transformation invariance of the dissimilarity functions used for matching polynomial curves are difficult to extend to closed active contours. In fact, splitting contours into the segments necessary for tracking heart walls would require segmentation criteria similar to those we use for contour segmentation. Polynomial curves offer simplicity and at the same time are sufficiently accurate to model heart walls.

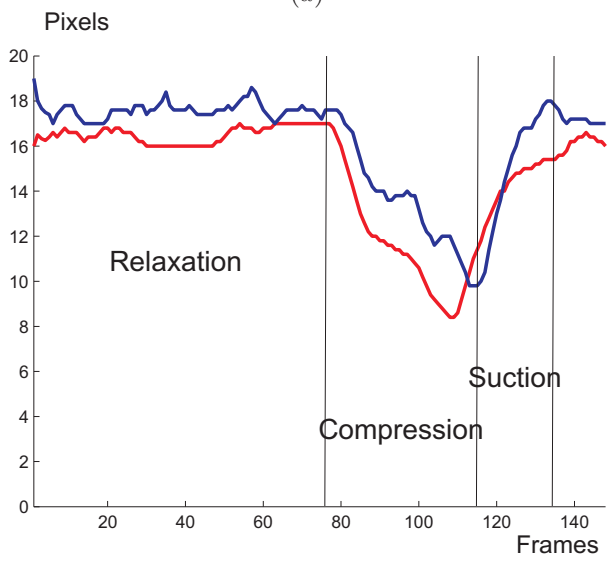
Modelling blood cells with circles using the CHT

The general idea of the Hough transform is to define a mapping from the preprocessed feature space to a parameter space, the so called accumulator, where each pixel that belongs to the contour of an object in the image space is mapped to the same point in the accumulator. The objects in the image are detected by post-processing local maxima in the accumulator. We use the CHT for greyscale images, based on the gradient field of an image. The range of radii is limited to $r = 6, 7, 8, 9$ pixels. The result of the CHT on a frame is shown in Figure 5.8(a). The circles on locations with no motion are excluded. This motion is obtained with Basic Background Subtraction.

The detection rate of cells is about 98.90%, this is an average of 90 detections over 91 frames, for each of five cells that we follow. Computed over 148 frames, the average percentages of true and false positives are 73.53% and 26.47%, respectively. The false positives are not a real problem for the estimation of the velocity. They can be excluded either because they have a match on the same location or because the correlation is too low.

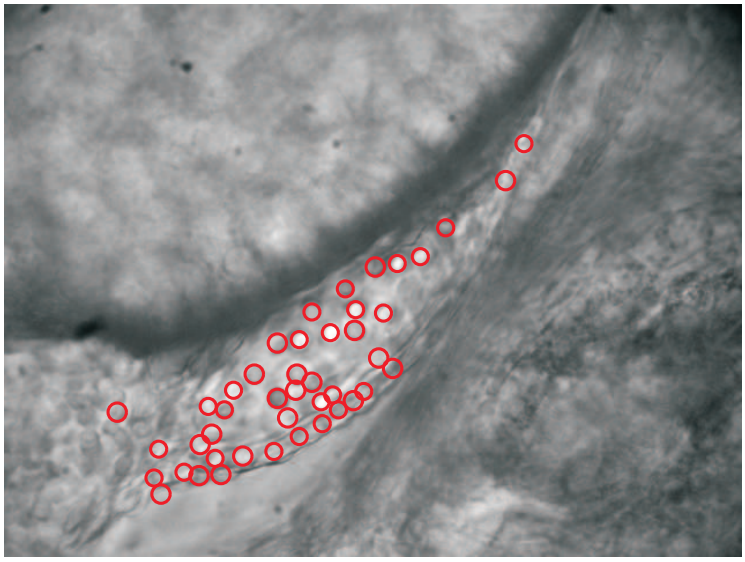


(a)

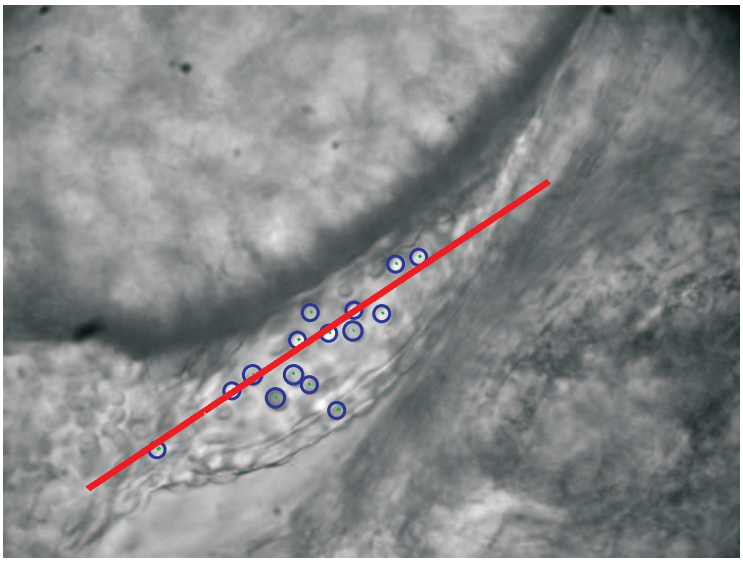


(b)

Figure 5.7: (a): The heart wall segmented using an active contour, indicated in red colour. We measure the positions of the heart wall on the blue line segment. (b): The blue and the red curve correspond to the position estimates of the heart wall and the ground truth, respectively.



(a)



(b)

Figure 5.8: (a): The result of the CHT. (b): The circles with their motion vectors that remain after matching. The motion vectors are projected on the red linesegment, which is a rough approximation for the moving direction in the fluid flow.

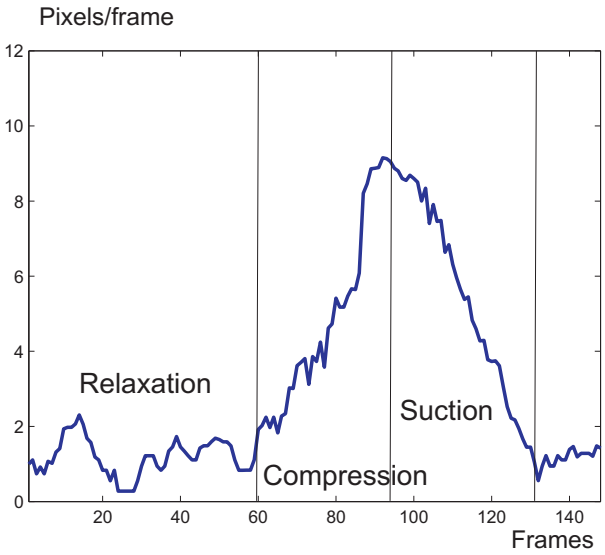


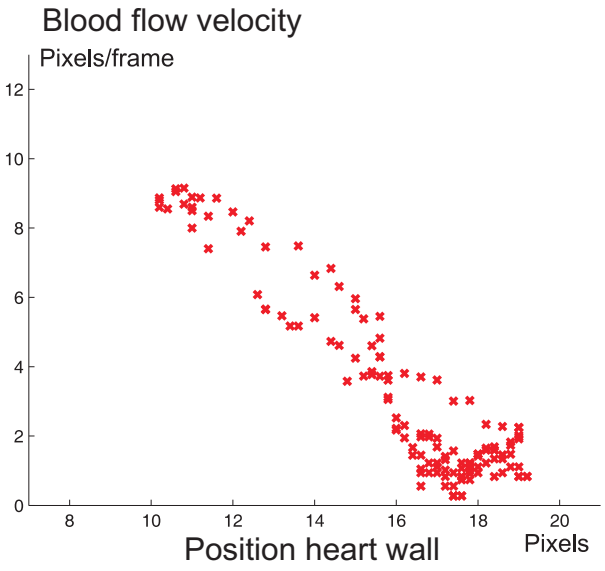
Figure 5.9: The velocity estimates in the fluid flow over 148 frames. The three phases of a pumping heart are indicated: the relaxation, the compression and the suction phase.

Tracking of circles with local intensity correlation

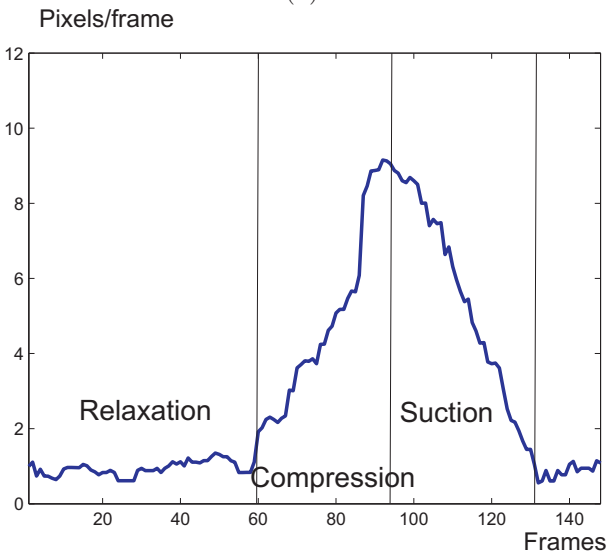
Since we have well-contrasted, spatially isolated cells of fixed shapes we can use intensity correlation for the matching of circles. Two circles A and B in consecutive frames have a match, if the Euclidean distance of their centres is lower then a radius R and if they have a maximized correlation coefficient higher then a threshold C, the remaining circles in the result are shown in Figure 5.8(b). Actual values for R and C are 15 pixels and 0.85. The size l of the square correlation window depends of the radius r_A of the circle A: $l = 2r_A + 1$.

The main direction of the fluid flow is roughly estimated from the histogram of directions in which the cells move. Projection of the motion vectors of the circles on this main direction gives us the spatial velocity of the fluid flow, as shown in Figure 5.8(b).

We define the success rate of cell tracking as the ratio of frames in which the cell is detected and tracked, to the total number of frames in the sequence. We have a tracking percentage of 90.11%, this is an average of 82 good tracks over 91 frames, for each of five cells that we follow.



(a)



(b)

Figure 5.10: (a): The correspondences between the positions of the heart wall and the velocities in the fluid flow. The velocities in the fluid flow are ten frames delayed. (b): The improved velocity estimates in the fluid flow.

Measurement of the velocity in the fluid flow

Figure 5.9 shows the velocities for one cycle of the fluid flow. In this curve, we can indicate the three phases of the pumping mechanism of the heart: the relaxation, the compression and the suction phase.

Improved fluid flow estimation

In the graphs of the velocities in the fluid flow and the positions of the heart wall, we can distinguish three similar phases of the pumping mechanism of the heart, namely the suction, the compression and the relaxation phase. The velocity in the fluid flow react with a delay of ten frames to a change in position of the heart wall. The correspondences over time between the positions of the heart wall and the velocities in the fluid flow are plotted in Figure 5.10 (a). We use these correspondences to achieve a significantly higher matching rate for circle matching by introducing a region of interest. In our results, circles in consecutive frames at times $t-1$ and t can match, if the Euclidean distance between their centres is in a certain range of radii R_t, where R_t is dependent to the average position P_t of the heart wall in the frame at time t.

$R_t = [0,5]$ if $\sum_{q=0}^{4} P(t-10-q)/5 \in [15,20]$ and

$R_t = [3,15]$ if $\sum_{q=0}^{4} P(t-10-q)/5 \in [5,15[$.

Through this simple criterion, the tracking percentage for circles increases to 95.60%. This is an average of 87 good tracks, when following five cells over 91 frames. The percentage of false matches is decreased with 55,5%. Figure 5.10 (b) shows the much more smoothed velocity estimates in the fluid flow when compared to the velocity estimates in Figure 5.9.

5.4 Water current tracking

In this section, we demonstrate that polynomial curve models are not only suited to model and track rigid objects, but also to registrate motion of non-rigid objects, such as water currents. Firstly, CEMs are extracted in two consecutive frames of water current. The method to extract CEM is thoroughly explained in Section 2.5. Figure 5.11 shows the segmentation of water current in a natural scene into polynomial curves. Then, the two CEMs of the scene are matched using local curve matching as described in Section 3.3.1. Figure 5.12 shows the matching of polynomial curves in two consecutive frames of water current. Corresponding polynomial curves are indicated by the same colour. As last step, moving water current is detected by a cluster algorithm for sets of comparable curve motion vectors, as described in Section 3.3.1.5. Figure 5.13 shows the motion registration

(a) (b)

Figure 5.11: (a): A frame of a water current sequence in a natural scene. (b): The segmentation of a frame of a water current sequence into polynomial curves.

(a) (b)

Figure 5.12: (a) and (b): Polynomial curve matching in two consecutive frames of a water current sequence. Corresponding polynomial curves are indicated by the same colour.

of water current. The moving and stationary polynomial curves are indicated by the colours red and green, respectively. The polynomial curves representing the water current are nicely separated from the other part of the scene.

5.5 Conclusion

In this chapter, we evaluate tracking of other objects than faces, such as vehicles, heart walls and water currents, based on fitting, matching and motion registration of polynomial curves. The result is robust tracking of rigid and non-rigid objects,

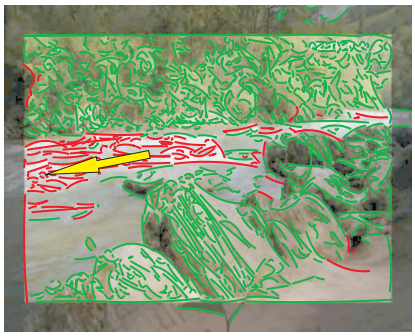


Figure 5.13: Motion registration in a water current scene. The moving and stationary polynomial curves are indicated by the colours red and green, respectively. The polynomial curves representing the water current are nicely separated from the other part of the scene.

which can cope with small changes in viewpoint on the moving object.

This research was published in two papers [Deboeverie et al., 2009a, Deboeverie et al., 2009b] in the proceedings of international conferences. This work is part of a contribution to [Maes et al., 2009].

6

Applications of segmented face approximation

6.1 Introduction

Segmented face approximation has its applications in visual communication. The objective in visual communication is to send and store images of faces at a low bit rate, such that the faces are still recognizable and that the compression does not prevent remote face analysis. For instance in automated video conferencing, intelligent cameras employ face analysis systems with on-board face databases. In this application, a compact face representation will reduce processing and storage cost. Ideally for video analysis, such a representation would be based on a perfect segmentation of a face into all its major parts, i.e., forehead, nose, eyes, lips, etc. For example, it is important that lip motion follows speech very closely with almost no time delay, which is much easier to accomplish when the image representation already contains the lips as separate items.

In this chapter, we approximate facial contour pixels and facial image intensities with maps of polynomial curves (CEMs) and maps of polynomial surfaces (SIMs), respectively. The polynomial representation provides good approximation of facial features, while preserving all the necessary details of the face in the reconstructed image. When compared with different compression methods, we achieve higher compression ratios and better recognizable faces at low bit rates. This is confirmed by correct identification percentages obtained by face recognition algorithms on the compressed data. Furthermore, the curvature-based surface shape

information facilitates many tasks in automated face analysis, demonstrated in this chapter by face verification performed on the polynomial representation.

The work in this chapter was published in [Deboeverie et al., 2013c].

This chapter is structured as follows: in Section 6.2, we discuss related work. In Section 6.3, we shortly explain the method of segmented face approximation. In Section 6.4, we thoroughly evaluate the proposed method.

6.2 Related work

We give an overview of the most recent approaches that use segmentation-based image approximation. Biswas [Biswas, 2003] proposed a segmentation based lossy image compression (SLIC) algorithm. The segmentation scheme recursively uses an object/ background thresholding algorithm based on conditional entropy. SLIC encodes images through approximation of segmented regions by 2-d Bezier-Bernstein polynomials, contours by 1-d Bezier-Bernstein polynomials (line and arc segments) and texture by a Huffman coding scheme using Hilbert scan on texture blocks. Other popular approaches to segmentation-based image approximation are curvilinear-based and region-based. Curvilinear-based approximation produces tree representations that specify the boundaries of regions. Some examples are curvelets [Candès and Donoho, 1999], wedgelets [Donoho, 1999], beamlets [Donoho and Huo, 2001], contourlets [Do and Vetterli, 2003], platelets [Willett and Nowak, 2003], bandelets [Pennec and Mallat, 2005] and geometric wavelets [Alani et al., 2007]. Region-based approximation relies on tree structures to describe the interior of the regions. It includes image quadtree decomposition and binary space partitioning. In image quadtree decomposition [Samet, 1984, Sullivan and Baker, 1994], each partition is recursively subdivided into more homogeneous quadrants. In binary space partitioning (BSP) [Radha et al., 1996, Shukla et al., 2005], the image plane is recursively bi-partitioned along arbitrarily-orientated linear boundaries, constrained by a least-squares-error measure or a contour-matching criterion. In the latter two techniques, the image data within each convex polygonal image region can be approximated by low-degree polynomials. More recently, Kassim et al. [Kassim et al., 2009] presented the quad-binary (QB) tree, which is a compromise between the rigidity of discrete space structures of quadtrees, which allows spatial partitioning for local analysis, and the generality of BSP tree, which facilitates the creation of more adaptive and accurate representations of image discontinuities.

A number of algorithms are available for image segmentation and compression, but relatively few algorithms consider the treatment of face images [Pappas, 1992, Li et al., 1993, Moghaddam and Pentland, 1995, Hu et al., 1996, Lanitis et al., 1997, Sakalli and Yan, 1998, Ruppertsberg et al., 1998, Lyons and Akamatsu, 1998, Bartlett et al., 2000, Gerek and Çinar, 2004, Vila-Forcén et al., 2006, Elad

et al., 2007, Bryt and Elad, 2008, Somasundarama and Palaniappan, 2011]. Among the segmentation methods, only a very few algorithms are dealing with the segmentation of face images into meaningful regions. Pappas et al. [Pappas, 1992] considered segmentation of face images into smooth surfaces using a generalization of the K-means clustering algorithm [Tou and Gonzalez, 1974]. Gerek et al. [Gerek and Çinar, 2004] presented segmentation of face images into face regions with vector quantization clustering [Yoo et al., 2002]. More general techniques to segment the face image into different regions are marker-controlled watersheds [Jackway, 1996], morphological operators [Bosworth and Acton, 2000, Salembier et al., 1996], region growing [Adams and Bischof, 1994, Yemez et al., 1997, Pohle and Toennies, 2001, Qin and Clausi, 2010, Kanga et al., 2012], genetic algorithms [Aravind et al., 2002] combined with gradient information, mean shift [Comaniciu and Meer, 2002], normalized cuts [Shi and Malik, 2000], power watersheds [Couprie et al., 2011] and many other strategies [Muñoz et al., 2003].

Among the compression methods, Elad et al. [Elad et al., 2007] compressed face images using vector quantization (VQ) [Gersho and Gray, 1995]. Bryt et al. [Bryt and Elad, 2008] compressed face images based on the K-SVD algorithm [Aharon et al., 2006]. Both techniques train dictionaries on predefined face image patches, and compress each new face image according to these dictionaries. An essential pre-processing stage for these methods is a face image alignment procedure, where the handled images are geometrically deformed into a canonical form, in which facial features are located at the same spatial locations. In contrast to above-mentioned techniques, our method does not need training and face alignment, which is advantageous in real-time face analysis applications.

6.3 Method

Greyscale face images are segmented into meaningful surface segments with an adaptive region growing algorithm based on low-degree polynomial fitting, as described in Section 2.6. We represent the grey values in surface segments as polynomial surfaces of zeroth, first or second degree, as described in Section 2.6.3. These polynomials are either flat, planar, convex, concave or behave like saddle surfaces, as described in Section 2.6.4. In order to obtain a complete face representation, the contours separating the surface segments are segmented into contour segments and represented by low-degree polynomial curves, as described in Section 2.5. The contour segments are represented by their straight, convex and concave parts, which are polynomial curves of zeroth, first or second degree, as described in Section 2.5.4.

We represent each coefficient of the polynomial curves and surfaces with a variable bit length by performing uniform quantization and Huffman entropy coding. Firstly, polynomial coefficients are quantized by truncation to four bytes in

fixed point notation, including ten bits for the mantissa bits, which we experimentally decided to be sufficient to capture the full range of possible polynomial coefficients (the quantization error is approximately zero). This means in fixed point notation that we have a range of $[-2^{31}/2^{10}, (2^{31}-1)/2^{10}]$. Then, we remove coding redundancy by applying a straight-forward Huffman table, such that the polynomial coefficients get a variable bit length. Huffman coding [Huffman, 1952, Gonzalez and Woods, 2001] is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol, where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. The coordinates of the end pixels of the polynomial curves are represented by the number of bits needed to capture the maximal image size. For instance twelve bits is sufficient to capture a maximal image size of 2^{12} pixels.

In this chapter, we show that surface and contour segmentation allows parametrizing a face by the coefficients of the polynomial curves and surfaces and the coordinates of the endpoints of the curves in a few hundred bytes. Transmitting these face parameters over the network is very efficient and the method preserves all the necessary details of the face in the reconstructed image.

Curvature-based surface classification, as described in Section 2.6.4, delivers relevant information to perform face analysis, demonstrated in this section by face verification performed on the polynomial representation. The task of face verification is to verify a claimed face detection by analysing a claimed image of the face. In our case, face verification is performed on the polynomial representation. It uses two simple statistics from the output of our face representation, namely the number of segments and the curvatures of the polynomial surfaces. To detect if a bounding box contains a face if we use the following heuristic: the number of segments is low (e.g. ≤ 20) and there are four large concave polynomial surfaces (e.g. ≥ 225 pixels) of the forehead, the cheeks and the chin in a cross configuration. An example of such a cross configuration is shown in Figure 6.1. In practice, four segments are in a cross configuration, if the connecting lines between their centres of gravity intersect each other in the middle (e.g. between 40 and 60 percent of the length of the lines) and at a certain angle (e.g. between 80 and 100 degrees).

We evaluate segmented face approximation on the AR face database [Martinez and Benavente, 1998] and different video sequences, such as the standard video test sequences *Salesman*, *Claire* and *Carphone*. For compression ratios between 10 and 65, we achieve a PSNR between 40dB and 30dB, like most of the lossy image compression techniques. We also demonstrate that face detection and recognition performs better on images reconstructed with our segmentation-based approximation than reconstructions with different compression methods, such as the JPEG2000 [Taubman and Marcellin, 2002], the VQ method [Elad et al., 2007],

Figure 6.1: The convex, concave or saddle like behaviour of the polynomial surfaces in a face image, indicated by the colours magenta, cyan and yellow, respectively. Four large concave polynomial surfaces of the forehead, the cheeks and the chin are in a cross configuration.

the K-SVD method [Bryt and Elad, 2008], the QB-tree method [Kassim et al., 2009] and an L_2 variant of our method. For low numbers of bytes, we achieve higher PSNR and better recognizable faces. This is confirmed by correct identification percentages on the approximated images obtained by the Viola and Jones face detector (VJ) [Viola and Jones, 2001] and face recognition using Principal Component Analysis (PCA) [Turk and Pentland, 1991]. Convincing results for face analysis based on the curvatures of the polynomial surfaces are demonstrated by face verification on faces found by the VJ face detector. This face detector is widely used. However, it produces many false positives. In order to reduce the fraction of false positives, we use segmented face analysis to verify if the bounding boxes produced by this face detector actually contain faces.

6.4 Evaluation

Segmented face approximation evaluation

In this paragraph, we evaluate the proposed segmented face approximation with polynomial surfaces and curves (PSC) on the AR face database [Martinez and Benavente, 1998]. The AR face database consist of two series of thirteen face images of 136 persons under various circumstances, with 76 males and 60 females. The face images have an image size of 192x144. The results presented in this

(a) (b)

Figure 6.2: (a) and (b): Approximated face images with PSC and JPEG2000, respectively. They both need an equal number of bytes. However, (a) is sharper than (b).

chapter are obtained by evaluating the entire database. Next, we compare our method with different compression methods, considering compression ratio, face detection and face recognition performance.

We compare our PSC representation with the JPEG2000 [Taubman and Marcellin, 2002], the VQ method [Elad et al., 2007], the K-SVD method [Bryt and Elad, 2008], the QB-tree method [Kassim et al., 2009] and an L_2 variant of our method, for faces of the AR face database. Under the same conditions of PSNR and the number of bytes, we consider face detection using the VJ face detector [Viola and Jones, 2001] and face recognition using PCA [Turk and Pentland, 1991] on the approximated images. Figures 6.2 (a) and (b) show approximated face images with PSC and JPEG2000, respectively. They both need an equal number of bytes. However, (a) is sharper than (b).

The graph in Figure 6.3 plots the compression ratio versus the PSNR. For low PSNR, which corresponds to low approximation accuracies, the compression ratios with PSC compression are higher than those produced by the techniques considered in our comparison. When examining the relationship between the number of bytes and image quality, we find that PSC compression outperforms existing techniques for low approximation accuracies.

The following results demonstrate that faces in coded face images are still recognizable. This is important for instance when we perform remote face analysis

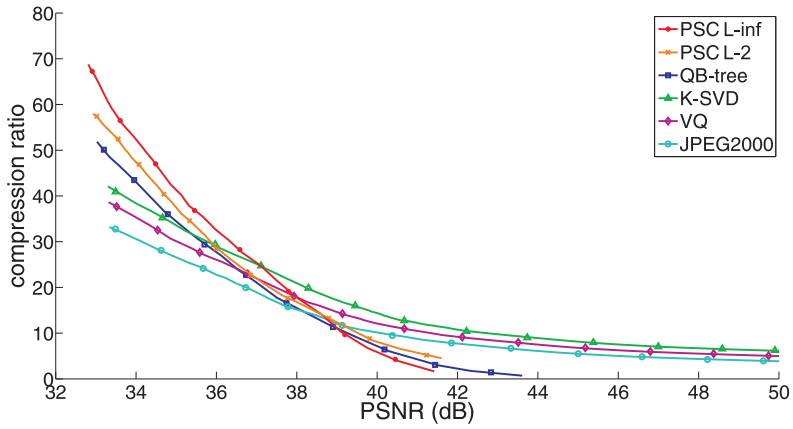
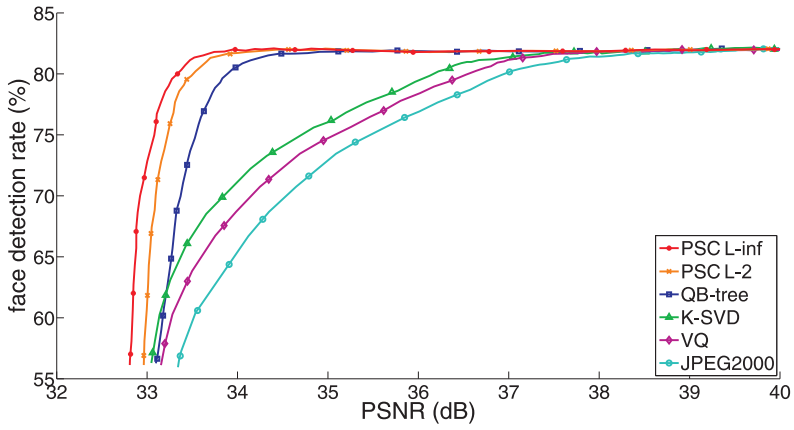


Figure 6.3: PSC compression is compared with the JPEG2000, the VQ method, the K-SVD method, the QB-tree method and an L_2 variant of our method, for faces of the AR face database. The graph plots the compression ratio versus the PSNR. When compared with existing techniques, PSC compression achieves better compression ratios for low approximation accuracies. Moreover, the L_∞ norm outperforms the L_2 norm.

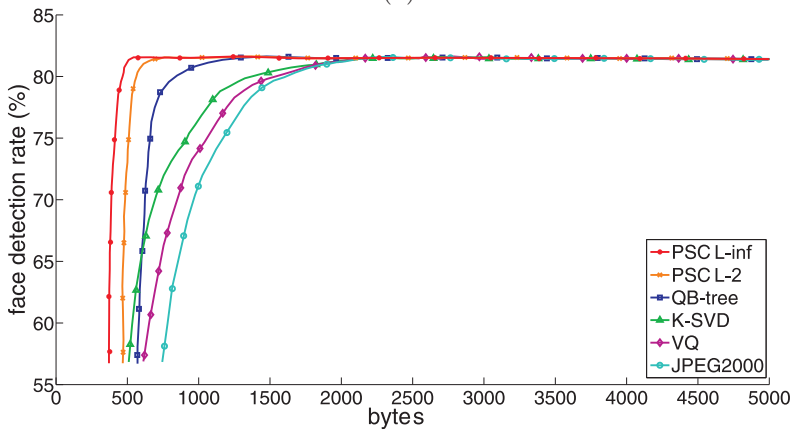
algorithms on coded face images in automated video conferencing.

The graphs in Figures 6.4 (a) and (b) plot the VJ face detection rate on approximated face images versus the PSNR and the number of bytes, respectively. At the core of this face detector is a cascade of complex classifiers. Each complex classifier consists of several simple classifiers that detect specific Haarlike features. An image region is classified as being a face if the region has passed all classification stages of the cascade. The graphs show that for low numbers of bytes as well as low PSNR, the face detection rates with PSC compression are higher than those of the techniques considered in our comparison.

The graphs in Figures 6.5 (a) and (b) plot the PCA-based face recognition rate on approximated face images versus the PSNR and the number of bytes, respectively. In this experiment of PCA-based face recognition, one series of thirteen available face images per person is used for training. A subset of ten eigenfaces (principal components) that correspond to the highest eigenvalues is selected to recognize face images. These eigenfaces define the face space (principal component space).

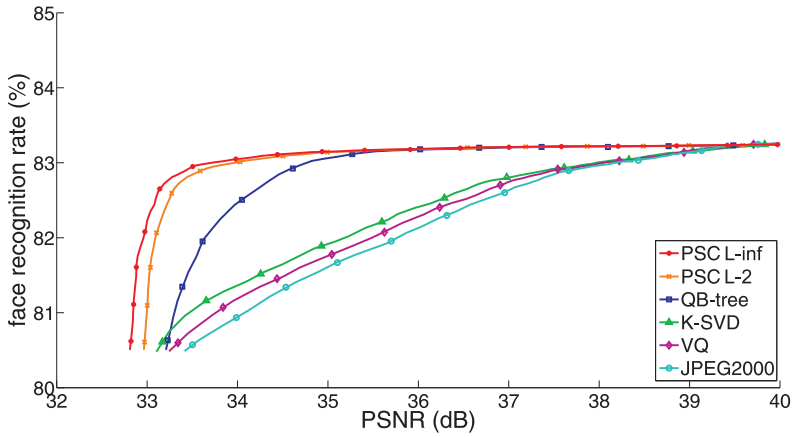


(a)

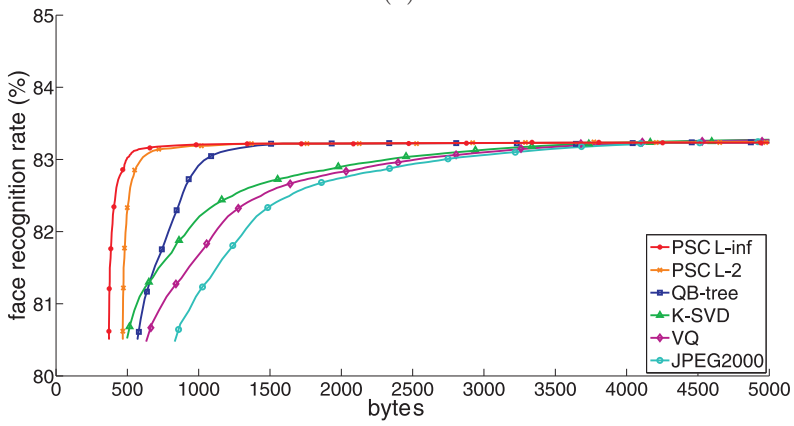


(b)

Figure 6.4: PSC compression is compared with the JPEG2000, the VQ method, the K-SVD method, the QB-tree method and an L_2 variant of our method, for faces of the AR face database. Graphs (a) and (b) plot the VJ face detection rate on approximated face images versus the PSNR and the number of bytes, respectively. When compared with existing techniques, PSC compression achieves better results for face detection for low numbers of bytes as well as low approximation accuracies. Moreover, the L_∞ norm outperforms the L_2 norm.



(a)



(b)

Figure 6.5: PSC compression is compared with the JPEG2000, the VQ method, the K-SVD method, the QB-tree method and an L_2 variant of our method, for faces of the AR face database. Graphs (a) and (b) plot the PCA face recognition rate on approximated face images versus the PSNR and the number of bytes, respectively. When compared with existing techniques, PSC compression achieves better results for face recognition for low numbers of bytes as well as low approximation accuracies. Moreover, the L_∞ norm outperforms the L_2 norm.

(a) (b) (c)

Figure 6.6: (a), (b) and (c): The convex, concave and saddle polynomial surfaces in segmented face images, indicated by the colours magenta, cyan and yellow, respectively.

For each known person, the training face images are projected into the face space to obtain the training eigenfeature vector of that person. Then, another series of thirteen available face images per person is first coded and then used for testing. Therefore, we project a test face image into the face space to get the test eigenfeature vector. Face recognition is performed by comparison of a test face image and a set of training face images of a person by calculating the Euclidean distance between the training eigenfeature vector and the test eigenfeature vector. We determine which person provides the best description of an input test face image by finding the set of training face images of the person that minimizes the Euclidean distance. The face recognition counts a correct identification when the best set of training face images is the correct person. The graphs show that for low numbers of bytes as well as low PSNR, the face recognition rates with PSC compression are higher than those of the techniques considered in our comparison.

When compared with existing techniques, PSC compression achieves better results for face detection and face recognition for low numbers of bytes as well as low approximation accuracies. In contrast to existing techniques, our approximated face images are better recognizable. When comparing the usage of the L_∞ and the L_2 fitting cost in an adaptive region growing process, we conclude that the L_∞ norm outperforms the L_2 norm.

Face verification evaluation

In this section, we demonstrate that curvature-based surface shape classification facilitates automated face analysis, demonstrated in by face verification performed on the polynomial representation. The method of face verification is explained in Section 6.3. Figures 6.6 (a), (b) and (c) show the concave, convex and saddle surface segments, indicated by the colours cyan, magenta and yellow, respectively. We find concave polynomial surfaces for the forehead, the cheeks, the chin and the nose, while the throat is often a saddle surface.

Face detection	TP	FN	FP
VJ without verification	946	379	235
VJ with PWS verification	946	379	78
VJ with PSC verification	946	379	17

Table 6.1: The results of TP, FN and FP for face detection performed on the video test sequences *Salesman*, *Claire* and *Carphone*. The total number of frames is 1325, of which each frame contains a face. Face verification is once performed with the segments produced by the PSC method as well as once with the segments produced by the power watersheds method (PWS) [Couprie et al., 2011]. Face verification drastically decreases the number of FP.

Figure 6.7: Three bounding boxes produced by the VJ face detector, of which two (blue ones) are rejected after face verification.

Face verification is performed on the bounding boxes produced by the VJ face detector. As a comparison, face verification is once performed with the segments produced by the PSC method as well as once with the segments produced by the power watersheds method (PWS) [Couprie et al., 2011]. Tests are performed on different video sequences, such as the standard video test sequences *Salesman*, *Claire* and *Carphone*. The total number of frames is 1325, of which each frame contains a face. Table 6.1 shows the results of True Positives (TP), False Negatives (FN) and False Positives (FP). Face verification significantly decreases the number of FP produced by the VJ face detector, which indicates that the proposed face verification is meaningful. However, the segments produced with the PSC method are more suited to perform face verification, since the number of FP decreases the most. Figure 6.7 shows three bounding boxes produced by the VJ face detector, of which two (blue ones) are rejected after face verification is performed. Because FP are discarded, face verification contributes to the total compression.

6.5 Conclusion

Segmented face approximation with polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image. Face images are represented by flat, planar, convex, concave and saddle polynomial surfaces with a variable fitting error. The boundaries are represented by straight, convex and concave curves. This way, we are able to represent recognizable faces using a few hundred bytes rather than a few hundred kilobytes, which is very useful in visual communication applications, such as automated video conferencing. Moreover, the polynomial surfaces correspond to meaningful facial features. When compared to existing approximation techniques, we achieve higher compression ratios and better recognizable faces. Furthermore, the polynomial surfaces are well suited to automated face analysis. Future research will focus on compression of the polynomial coefficients by entropy minimization and huffman encoding.

This research was published in an international journal [Deboeverie et al., 2013c].

7 Human body analysis

7.1 Introduction

Human body segmentation divides images of human bodies into parts that coincide with limbs, like torso, arms or legs. Segmenting human body images is a first step to computer vision-based analysis of human movements patterns [Aggarwal and Ryoo, 2012]. Furthermore, it facilitates many useful applications such as surveillance, human action understanding, pose classification, etc. [Juang et al., 2009, Liang et al., 2009, Hou and Pang, 2011]. Reliable segmentation leads to good qualitative movement analysis, for instance in physiotherapy. Movement analysis helps athletes, for instance gymnasts, to improve their performance and to reduce the risk of injury [Bartlett, 2007]. Analysis of human movements is often performed by finding human body configurations in human body skeleton reconstructions [Mori et al., 2004]. However, in order to accurately reconstruct these human body skeletons, human body parts segmentation is still challenging.

In this chapter, greyscale images of human bodies are segmented into smooth surface segments and then approximated with maps of polynomial surfaces (SIMs). These human body parts are approximated by nearly cylindrical surfaces, of which the axes of minimum curvature accurately reconstruct the human body skeleton. For the reconstructed human body skeleton, the branches generally coincide with the real human body bones, because the cylindrical surfaces have the same shape as the limbs. Human body segmentation is qualitatively evaluated with a line segment distance between reconstructed human body skeletons and ground truth

skeletons. When compared with human body segmentations based on mean shift, segmentations based on normalized cuts and segmentations based on watersheds, our method achieves more accurate segmentations and better reconstructions of human body skeletons.

The work in this chapter has been submitted to an international journal [Deboeverie et al., 2013a].

This chapter is structured as follows: in Section 7.2, we discuss related work. In Section 7.3, we shortly explain the method of segmented face approximation. In Section 7.4, we thoroughly evaluate the proposed method.

7.2 Related work

Many methods related to the segmentation of a human body into its different parts have been proposed by researchers. The most recent papers consider human body segmentation in *videos* [Juang et al., 2009, Hsieh et al., 2010, Liu et al., 2011a, Shao et al., 2012] as well as in *static images* [Cour and Shi, 2007, Srinivasan and Shi, 2007, Barnard and Heikkila, 2008, Li et al., 2011]. For videos, the methods are performed on human bodies which are segmented by using a background model and motion information, which make them infeasible for static images. The methods for static images are further classified into matching-based [Mori and Malik, 2002, Mori and Malik, 2006], part-based [Mori et al., 2004, Hu et al., 2006] and model-based [Lee and Cohen, 2006, Hu et al., 2009] methods. Matching-based methods compare human body features (such as shape contexts [Mori and Malik, 2006]) in a test image with those in a large set of labelled images. Part-based approaches detect the candidates of each body part (such as torso and limbs) and construct the best assembly according to some predefined human body configuration constraints. Model-based methods firstly generate a large number of hypotheses of human body configurations and then recover the human body configuration by minimizing the errors between the hypotheses and the image.

Different from these methods, the approach in this work does not require a training test scheme or a model hypothesis on the human body. Furthermore, none of these techniques consider a segment curvature approach for segmentation. As we will show in the results, considering the curvature in the segmentation process enhances the reconstruction of human body skeletons.

7.3 Method

Greyscale human body images are segmented with curvature-based segmentation, as described in Section 2.6. Curvature-based segmentation finds surface segments that are folded in a certain way. The grey values in surface segments are repre-

sented as polynomial surfaces of zeroth, first or second degree, as described in Section 2.6.3. These polynomials represent either flat, planar, convex, concave or saddle surfaces, as described in Section 2.6.4. Furthermore, when examining the Gaussian curvature (Section 2.6.4) of second-degree polynomials, we find that many human body parts are represented as nearly cylindrical surfaces. For these cylindrical surfaces, the axes of minimum curvature accurately reconstruct the human body skeleton.

Human body segmentation is evaluated on images of an athlete performing different exercises. As a measure for segmentation quality, we examine a line segment distance [Gao and Leung, 2002a] between human body skeleton reconstructions and ground truth skeletons. Considering this quality measure, we find that the proposed segmentation outperforms human body segmentation techniques based on mean shift [Comaniciu and Meer, 2002, Porikli and Tuzel, 2003], segmentations based on normalized cuts [Shi and Malik, 2000, Mori et al., 2004, Li et al., 2011] and segmentations based on marker-controlled watersheds [Vincent and Soille, 1991, Beucher and Meyer, 1993, Beucher, 1994, Jackway, 1996, Park et al., 1999].

7.4 Evaluation

Experimental video data includes an athlete performing 10 different exercises. Figure 7.1 (a) shows a greyscale human body image of the athlete with image size 130x180. As usual in physiotherapy, the athlete only wears short pants. The availability of a diffuse reflecting skin surface makes these images very suitable to test the segmentation method on. Ground truth skeleton data of the human body is obtained from markers which are labelled on the joints of the athlete by physiotherapists. An example of a ground truth skeleton is shown in Figure 7.1 (b). This ground truth data is used to evaluate the method for human body segmentation. Figure 7.1 (c) shows a first segmentation result of a human body image. The image is segmented into 40 surface segments. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. We ascertain that many surface segments correspond to meaningful parts of the human body, such as the arms, the legs and the torso. Figure 7.1 (d) shows the best fit polynomial surfaces approximating the surface segments. The result is a reconstructed image of the original image. The human body parts are nicely reconstructed from the low-degree polynomial surfaces. An example of convex, concave or saddle like behaviour is shown in Figure 7.1 (e). The magenta, cyan and yellow colours correspond to convex, concave and saddle like behaviour, respectively. We find concave polynomial surfaces for the arms, the legs and the torso. We experimentally found that human body parts which are shadowed by other human body parts are often approximated by convex polynomial surfaces.

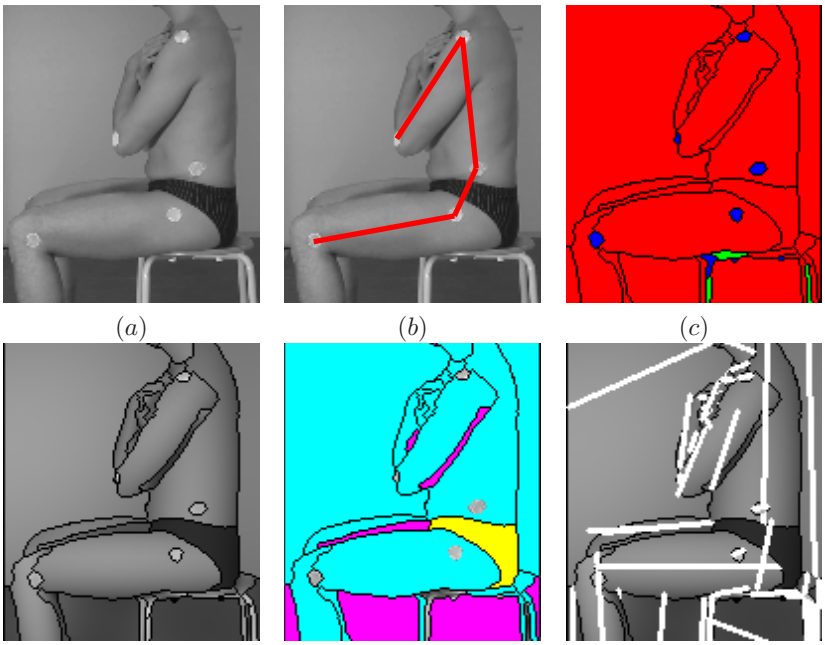


Figure 7.1: (a): A greyscale human body image with image size 130x180. (b): The ground truth human body skeleton. (c): The human body image is segmented into 40 surface segments. The blue, green and red colours in the segmented image correspond to zero, first and second degree polynomial surfaces, respectively. (d): The surface reconstructed human body image. (e): The convex, concave or saddle like behaviour of the polynomial surfaces, indicated by the colours magenta, cyan and yellow, respectively. (f): The axes of minimum curvature of the nearly cylindrical surfaces approximating the human body parts.

Figure 7.1 (f) shows the axes of minimum curvature of the polynomial surfaces. These axes go through the center points of the segments. For the human body segments, these axes coincide with the real human body bones. Together, they form a reconstructed human body skeleton. In these Figures, the columns (a), (b), (c) and (d) show the original greyscale images, the segmented images, the reconstructed images and the images with the axes of minimum curvature of the polynomial surfaces, respectively. In the images in column (b), the surface segments of the human body are separated from the background by considering the segments close to the markers.

To find the optimal parameter set of adaptive region growing, we measure the image approximation accuracy with a surface area weighted mean of the L_∞ fit-

Performance statistics	
image size	180x130
mean fitting cost	3.97 ± 0.42
#surfaces	17.83 ± 3.14

Table 7.1: This table always shows the mean and standard deviation of the mean fitting cost and the number of surfaces, respectively. Only very few surface segments are needed to represent a human body image.

ting costs of the polynomial surfaces . A high approximation accuracy (low mean fitting cost) leads to a high number of smaller segments, providing a good approximation quality. On the other hand, a low approximation accuracy (high mean fitting cost) leads to a low number of larger segments, providing approximation quality less well. Depending on the desired purpose (approximation or segmentation), one has to find a good balance between the size of the segments and the quality of the approximated images. For the results in this work, we set the segmentation parameters $T_X = 0.8$ and $T_Y = 4.8$, preserving a good size of the segments to perform analysis. These parameters have been manually tuned on a small number of images. When considering a set of 200 human body images, Table 7.1 always shows the mean and standard deviation of the mean fitting cost and the number of surfaces, respectively. We find that our technique divides a human body image in only very few surface segments.

The graph in Figure 7.2 plots the numbers of surface segments when segmenting human body images in function of different mean fitting costs (7.4). We find that for mean fitting costs above 5, which corresponds to low reconstruction accuracies, the mean numbers of surface segments remains more or less constant. This means that there is a small stable set of large segments. In contrast, for mean fitting costs below 5, which corresponds to high reconstruction accuracies, the mean number of surface segments grows exponentially in function of the mean fitting cost. This means that there are many small segments.

Figure 7.3 shows the axes of minimum curvature of the second-degree polynomial surfaces approximating the human body parts in Figure 7.1 (e). For these surfaces, Table 7.2 always shows the minimum and maximum curvatures, the Gaussian curvatures (Section 2.6.4) and the azimuths of minimum curvatures, respectively. From these values, we conclude that the polynomial surfaces approximating human body parts are nearly cylindrical, since the Gaussian curvature is zero when one of the eigenvalues is zero. The corresponding axes of minimum curvature reconstruct the human body skeleton. They coincide with the ground truth human body skeleton in Figure 7.1 (b).

In order to qualitatively evaluate human body segmentation, we match the reconstructed human body skeleton to the ground truth skeleton data with a Line

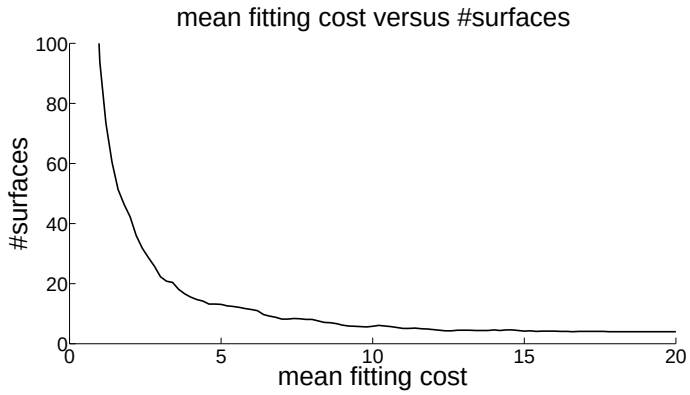


Figure 7.2: The numbers of surface segments of human body images in function of mean fitting costs.

Figure 7.3: This figure shows the axes of minimum curvature of the nearly cylindrical surfaces approximating the human body parts. When matched to the ground truth skeleton, the LSD is 3.90.

Segment Distance (LSD) as proposed in [Gao and Leung, 2002a]. The LSD is useful in skeleton matching, because it encourages one-to-one mapping of similar lines. Given two sets of line segments $M = \{m_1, m_2, \ldots, m_p\}$ and $T = \{t_1, t_2, \ldots, t_q\}$, the distance between two line segments m_i and t_j is defined as

$$d(m_i, t_j) = \sqrt{d_\theta^2(m_i, t_j) + d_\parallel^2(m_i, t_j) + d_\perp^2(m_i, t_j)} \tag{7.1}$$

where $d_\theta(m_i, t_j)$is the orientation distance, $d_\parallel(m_i, t_j)$ is the parallel distance and $d_\perp(m_i, t_j)$ is the perpendicular distance between m_i and t_j [Gao and Leung, 2002a]. The line segments m_i and t_j form a corresponding pair if $d(m_i, t_j)$ is a minimum over all combinations of t_j. From the pairs of matching line segments,

	λ_1	λ_2	G	θ
lower leg	-0.0063	-0.3668	0.0044	81
upper leg	-0.0020	-0.1750	0.0004	-2
lower arm	-0.0058	-2.0147	0.1907	-75
upper arm	-0.0150	-0.1815	0.0076	-53
torso	-0.0054	-0.1184	0.0007	-87

Table 7.2: This table always shows the minimum and maximum curvatures, the Gaussian curvatures and the azimuths of minimum curvatures of the second-degree polynomial surfaces approximating the human body parts in Figure 7.1 (e).

a length weighted matching cost $C(M,T)$ is computed:

$$C(M,T) = \frac{1}{\sum_{m_i \in M} l_{m_i}} \sum_{m_i \in M} l_{m_i} \min_{t_j \in T} d(m_i, t_j), \tag{7.2}$$

where l_{m_i} is the length of line segment m_i. This matching cost is used as a quality measure for human body segmentation, where a lower LSD corresponds to a better segmentation quality. For instance, matching the axes in Figure 7.1 (f) to the ground truth skeleton in Figure 7.1 (b) results in a LSD of 3.90.

To compare the proposed method with existing techniques that do not provide a reconstructed human body skeleton from curvature information of the surface segments, we also reconstruct the human body skeleton from the axes of an ellipse model for the surface segments. The difference between human body skeleton reconstruction from cylinders and ellipses is shown in the block diagrams in Figure 7.4. We obtain ellipses from least-squares fits. An ellipse model for human body parts was earlier proposed in [Park and Aggarwal, 2004, Wu and Aghajan, 2007]. An example human body surface segments represented by ellipses and their axes is shown in Figure 7.5 (a). To measure the quality of a human body segmentation, we match the axes of the ellipses with the ground truth skeleton using the LSD. Figure 7.5 (c) shows the axes of the ellipses that match with the ground truth skeleton in Figure 7.5 (b). The corresponding line segment pairs are indicated by the same colour. Here, the LSD is 6.34. The correct axes are matched, they follow almost the same directions as the groundtruth skeleton. Additional results are shown in Figure 7.6. The rows (a) to (f) show the input images, the segmented images, the surface reconstructed images, the reconstructed human body skeletons from the axes of minimum curvature, the ground truth human body skeletons and the axes of the ellipses representing the segments that match with the ground truth skeletons, respectively. The corresponding line segment pairs are indicated in the same colour. The surface segments of the human bodies are separated from the background by considering the segments close to the markers.

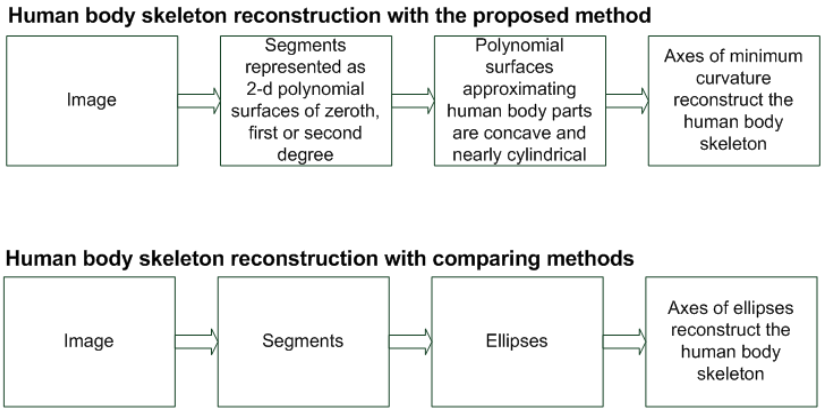
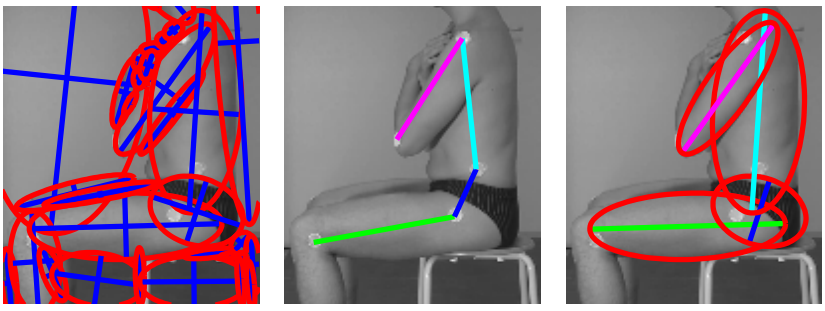


Figure 7.4: Two block diagrams which indicate the difference between human body skeleton reconstruction from cylinders and ellipses. The reconstruction from the axes of minimum curvature of cylinders only applies to the proposed method. In order to compare, the reconstruction from the axes of ellipses applies to methods that do not provide curvature information of the surface segments.



(a) (b) (c)

Figure 7.5: (a): The human body image segments represented by ellipses and their axes. (b): The ground truth human body skeleton. (c): The axes of the ellipses that match with the ground truth skeleton. The corresponding line segment pairs are indicated by the same colour. The LSD is 6.34.

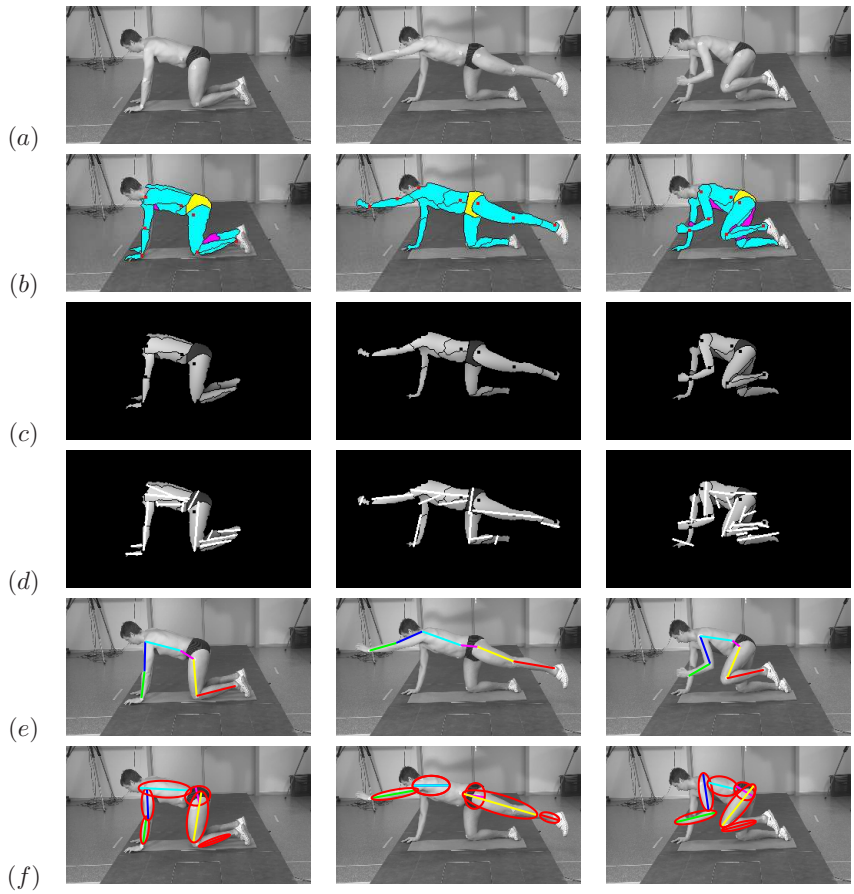


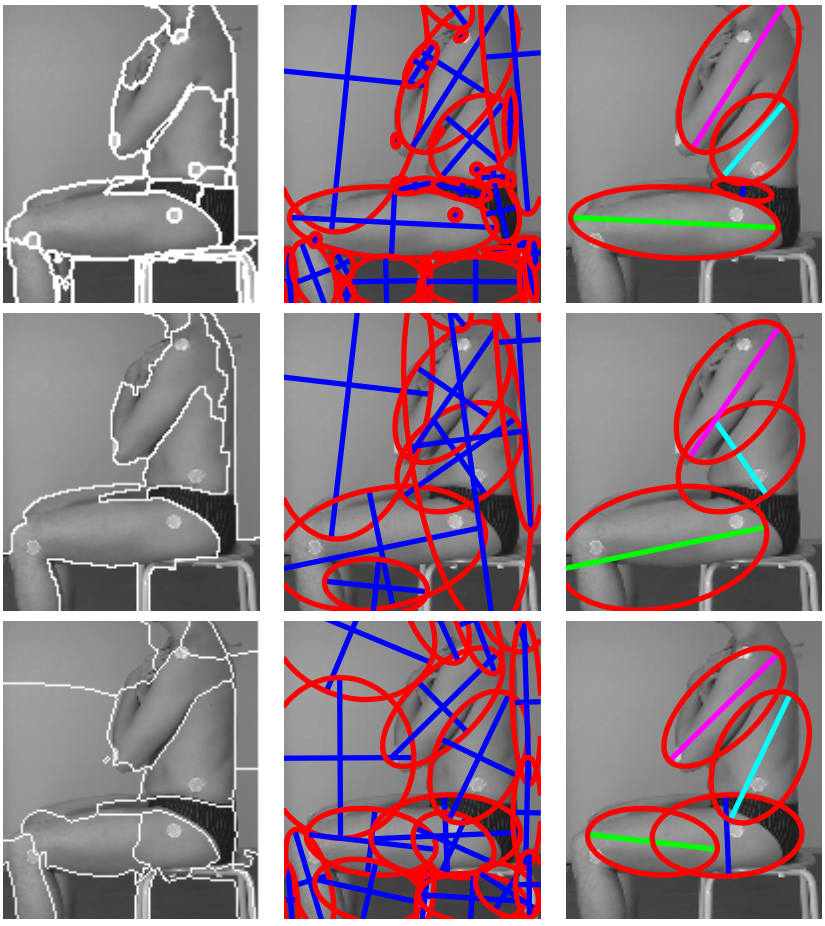
Figure 7.6: (a): Input images. (b): The segmented images. Many surface segments correspond to meaningful parts of the human body, such as the arms, the legs and the torso. The magenta, cyan and yellow colours correspond to convex, concave and saddle like polynomial surfaces. We find concave polynomial surfaces for the arms, the legs and the torso, while shadowed human body parts are often approximated by convex polynomial surfaces. (c): The surface reconstructed human body images. (d): Reconstructed human body skeletons from the axes of minimum curvature of the polynomial surfaces approximating the human body. (e): The ground truth human body skeletons. (f): The axes of the ellipses representing the segments that match with the groundtruth skeletons. The corresponding line segment pairs are indicated by the same colour.

Line Segment Distance	(lower is better)
Polynomial surfaces (cylinders)	4.87 ± 1.21
Polynomial surfaces (ellipses)	6.31 ± 2.17
Mean-shift	10.73 ± 2.99
Normalized cuts	11.64 ± 3.25
Marker-controlled watersheds	16.37 ± 3.28

Table 7.3: This table always shows the means and standard deviations of the LSDs for segmentation with polynomial surfaces, mean shift, normalized cuts and marker-controlled watersheds, respectively. For our method, beside the human body skeleton reconstructions from the axes of cylinders, we also make a variant of our method which produces human body skeleton reconstructions from the axes of ellipses.

We compare our segmentation based on polynomial surfaces, with human body segmentation algorithms based on mean shift [Comaniciu and Meer, 2002, Porikli and Tuzel, 2003], segmentations based on normalized cuts [Shi and Malik, 2000, Mori et al., 2004, Li et al., 2011] and segmentations based on marker-controlled watersheds [Vincent and Soille, 1991, Beucher and Meyer, 1993, Beucher, 1994, Jackway, 1996, Park et al., 1999]. Output examples of human body segmentation with mean shift, normalized cuts and marker-controlled watersheds are shown in column (a) in Figure 7.7. Column (b) in Figure 7.7 shows the segments represented by ellipses and their axes. Column (c) in Figure 7.7 shows the axes of the ellipses that match with the ground truth skeleton using the LSD. The LSDs for human body segmentation with mean shift, normalized cuts and watersheds are 10.56, 12.14 and 17.61, respectively.

Table 7.3 always shows the mean and standard deviation of the LSDs of human body segmentations based on polynomial surfaces, mean shift, normalized cuts and marker-controlled watersheds, respectively. For our method, beside the human body skeleton reconstructions from the axes of cylinders, we also make a variant of our method which produces human body skeleton reconstructions from the axes of ellipses. We find that the LSDs of other segmentation methods are higher than the LSD of our segmentation. Furthermore, a human body skeleton reconstruction with the axes of cylinders is more accurate than a reconstruction with the axes of ellipses. When we consider the means of the LSDs for segmentation of individual human body parts in Table 7.4, we see that our method outperforms the other techniques in all cases, especially for the skeleton reconstruction of the torso.



(a) (b) (c)

Figure 7.7: (a): The segmentation of a human body image with mean shift, normalized cuts and marker-controlled watersheds, respectively. (b): The segments represented by ellipses and their axes. (c): The axes of the ellipses that match with the ground truth skeleton. The LSDs are 10.56, 12.14 and 17.61, respectively.

LSD (lower is better)	lower leg	upper leg	lower arm	upper arm	torso
Polynomial surfaces (cylinders)	1.94	1.81	2.26	1.13	4.12
Polynomial surfaces (ellipses)	2.56	2.48	2.88	1.81	7.55
Mean-shift	4.17	3.64	4.05	2.83	19.83
Normalized cuts	4.69	3.25	4.54	2.21	27.33
Marker-controlled watersheds	5.03	4.86	5.39	2.96	24.84

Table 7.4: This table always shows the means of the LSDs for segmentation of individual human body parts with polynomial surfaces, mean shift, normalized cuts and marker-controlled watersheds, respectively.

Figure 7.8 shows the segmentation of an athlete bowing his torso. The atlete bows his torso once in a correct way and once in an incorrect way in the images on the second and the third row, respectively. The different way of bowing is clearly visible by the red line indicated on the back of the athlete. By sharing this information with the physiotherapist and the athlete, the athlete can improve the way he is performing excercises in order to obtain optimal sports or recovery results. We present additional results of human body segmentation of an athlete raising his arm, raising his knee, stretching his leg and jumping in Figures 7.9, 7.10, 7.11 and 7.12, respectively. In these Figures, the columns (a), (b), (c) and (d) show the original greyscale images, the segmented images, the reconstructed images and the images with the axes of minimum curvature of the polynomial surfaces, respectively. These experiments show that the curvatures of the surfce segments are valuable information to reconstruct the human body skeleton.

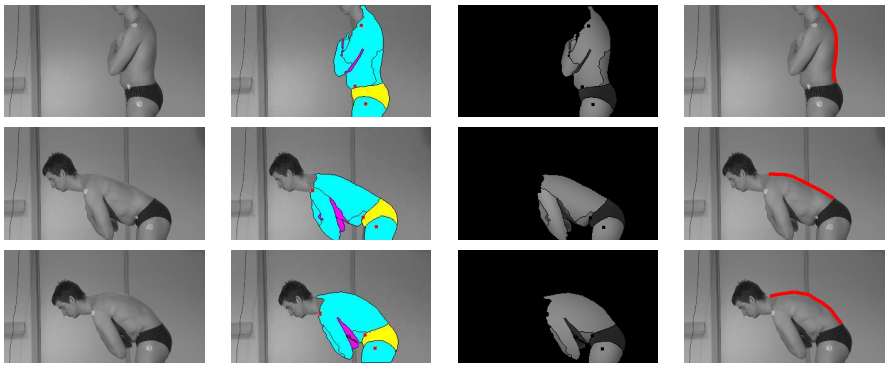
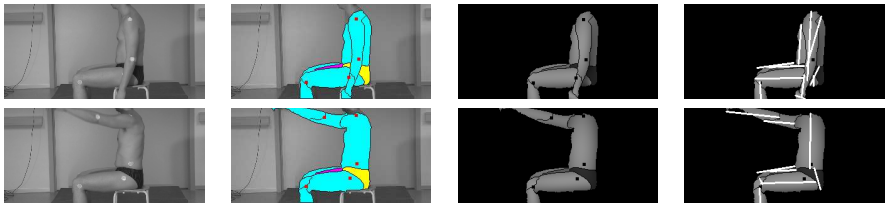
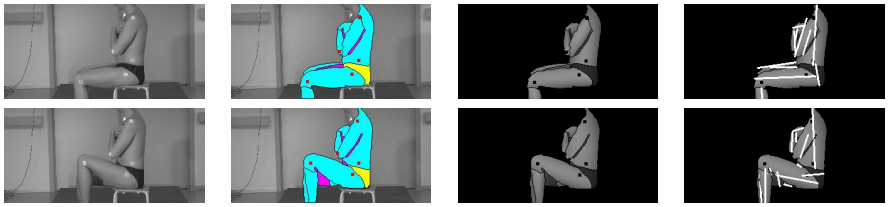


Figure 7.8: Segmentation of an athlete bowing his torso. The atlete bows his torso once in a correct way and once in an incorrect way in the images on the second and the third row, respectively. The different way of bowing is clearly visible by the red line indicated on the back of the athlete in the images in the last column.



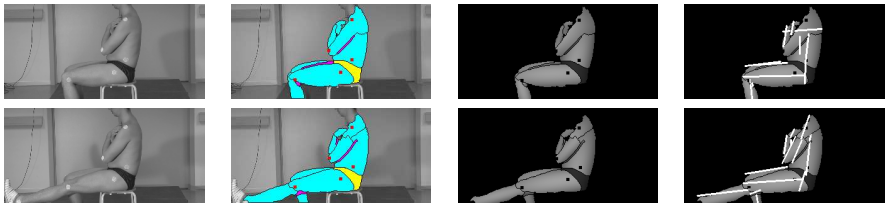
(a) (b) (c) (d)

Figure 7.9: Segmentation of an athlete raising his arm.



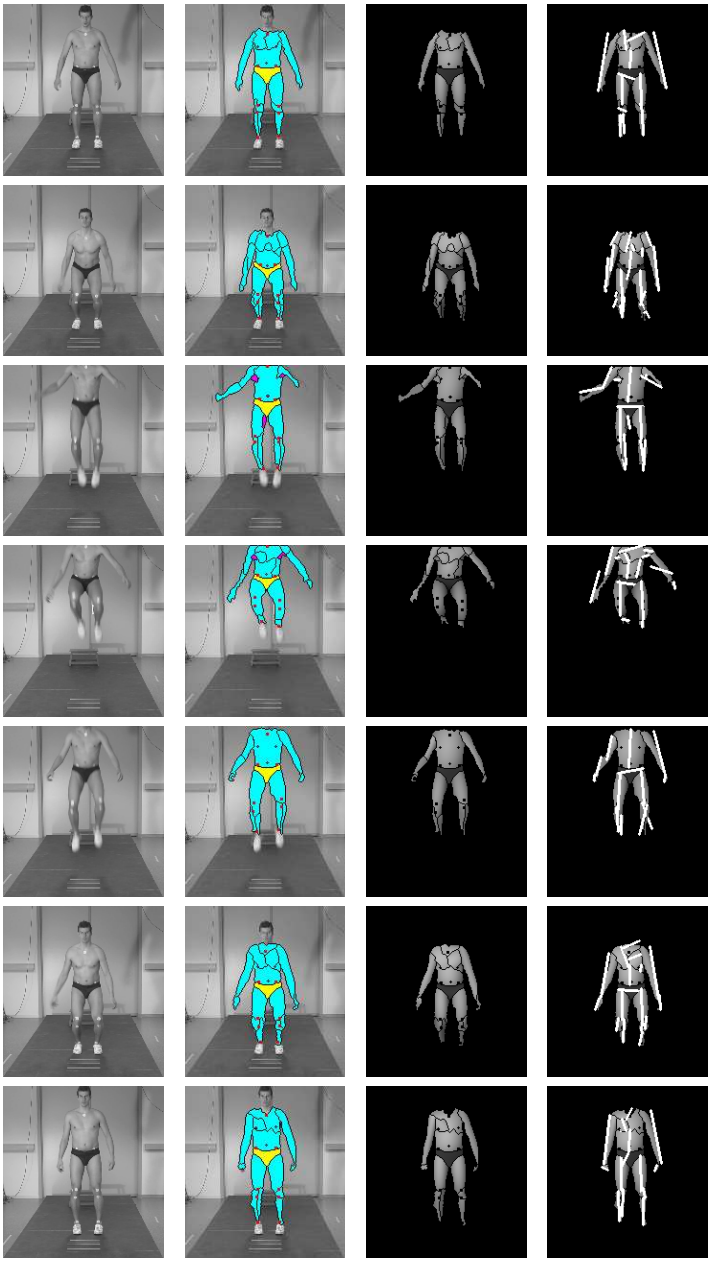
(a) (b) (c) (d)

Figure 7.10: Segmentation of an athlete raising his knee.



(a) (b) (c) (d)

Figure 7.11: Segmentation of an athlete stretching his leg.



(a) (b) (c) (d)

Figure 7.12: Segmentation of a jumping athlete

7.5 Conclusion

In this work, we perform human sports activity analysis with computer vision by reliable segmentation of the human body into meaningful parts, such as arms, torso and legs. Human body segmentation is performed on greyscale human body images by adaptive region growing based on constructive polynomial fitting. Human body images are represented by flat, planar, convex, concave and saddle polynomial surfaces with a variable fitting error. The low-degree polynomial surfaces correspond to meaningful human body features. We find that human body parts are often represented as nearly cylindrical surfaces, of which the axes of minimum curvature reconstruct the human body skeleton accurately. The proposed method, which provides human body skeleton reconstructions from curvature information of the surface segments, outperforms existing segmentation techniques that do not provide this information.

This research has been submitted in an international journal [Deboeverie et al., 2013a].

8 Conclusion

8.1 Overview

In this thesis, we deployed computer vision techniques to solve important problems about recognizing human identity and human behaviour in visual communication applications, such as automated video conferencing. We developed a system for face segmentation, approximation and analysis with polynomial contour and surface face models.

Face sementation is the division of face images into physically meaningful parts, such as the forehead, the cheeks, the lips, the eyebrows, etc. In Chapter 2, we proposed to group contour pixels and grey values in images of faces with region growing and polynomial fitting. Region growing is the process of examining neighboring pixels of initial seed pixels and determining whether the neighboring pixels should be added to the region. Polynomial fitting fits geometric low-level features, so-called geometric primitives, to pixels. A geometric primitive is a polynomial function describing the geometry of an edge or the variation of grey values in a region. Thus, the problem we studied is that of finding a region of maximal size in which grey values can be well approximated by a polynomial function and where contour pixels can be well approximated by polynomials as well. We considered the grouping of facial contour pixels into contour segments, as well as the grouping of facial intensities into surface segments.

To find segments, we proposed an adaptive region growing algorithm based on constructive polynomial fitting. This primitive extraction algorithm finds sub-

sets of pixels that lie on a geometric primitive or close to it. How well a subset corresponds to a primitive is measured by an L_∞ fitting cost (approximation error). This fitting cost is computed without computing the best fitting polynomial. The best fit is only computed when the segment is finished. An original contribution is the introduction of adaptive thresholding for region growing, which allows a variable polynomial degree and a variable fitting error, depending on the local properties of the pixels. The novelty is that region growing detects outliers, distinguishes between strong and smooth discontinuities and considers the curvature, such as convexity or concavity. The region growing investigates the local variation of pixels in a segment to identify outliers, while the global variation of pixels in a segment is investigated to adapt the degree of the polynomial function. The combination of both is possible because we employ constructive fitting: the global fitting cost is calculated from local fitting costs. In this work, we demonstrated how local and global fitting costs can interact in an adaptive method.

Face approximation is the estimation and the reconstruction of contour pixels and grey values of face images to a desired degree of accuracy. In Chapter 2, we proposed to approximate faces with maps defined by polynomial functions (geometric low-level features, geometric primitives) of low-degree (e.g., 0, 1 or 2). One difficulty we studied is that of finding the fitting parameters of the best fit.

The contour model represents contour segments as polynomial curves, which are either straight, convex or concave, in a Curve Edge Map (CEM). The surface model represents surface segments as polynomial surfaces, which are either flat, planar, convex, concave or saddle surfaces in a Surface Intensity Map (SIM). Both models are simple, natural, useful and elegant representations for objects in images, in particular for face images.

Segmented face approximation with polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image. Face images are represented by flat, planar, convex, concave and saddle polynomial surfaces with a variable fitting error. The boundaries are represented by straight, convex and concave curves. The low-degree polynomial functions in these models provide good approximation of meaningful facial features, while preserving all the necessary details of the face in the reconstructed image.

The contour and surface models allow to parametrizing a face by the coefficients of the polynomial curves and surfaces and the coordinates of the endpoints of the curves in a few hundred bytes. Transmitting these face parameters over the network is very efficient. Furthermore, these face descriptors are suitable for automated face analysis.

Face analysis includes recognition and tracking of faces and facial features. Face recognition identifies a person from a digital image or a video frame. Face tracking follows the movements of a person's head in a video. Recognition and

tracking of faces and facial features leads to the detection of specific human face behaviour, such as speaking, gaze direction, head movements (e.g. nodding) or facial expressions (e.g. happy/sad/angry/surprised).

In order to compare faces in two different or consecutive images, we presented a technique to find geometric feature correspondence pairs. The goal of a correspondence finding or matching algorithm is to indicate for a point (feature) in one image which is the corresponding point (feature) in a second image, where both image points must show the same 3D world point.

In the contour model in Chapter 3, we proposed to find correspondences by a technique that matches polynomial curves, based on shape, relative position and intensity. We proposed a dissimilarity function for local curve matching as well as a similarity function for global curve matching. The difference lies in the application: local matching is especially used in object tracking applications, while global matching focuses on object recognition applications.

In Chapter 4, we evaluated the matching techniques for polynomial curves by face recognition and tracking. We considered security applications, such as people identification and best view selection, and behaviour analysis applications, such as entering/leaving detection, head movement detection and speaker detection. We evaluated the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compared the proposed methods with several techniques of the state of the art. The face analysis results are comparable with or better than existing methods. The advantages of our methods are simplicity, real-time properties and many face analysis tasks are handled by the same approach.

In Chapter 5, we evaluated the matching techniques for polynomial curves by tracking of other objects than faces, such as vehicles, heart walls and water currents. The result is robust tracking of rigid and non-rigid objects, which can cope with small changes in viewpoint on the moving object.

In the surface model, we perform curvature-based surface shape analysis, as described in Chapter 2. The curvatures of polynomial surfaces roughly classify facial features into flat, planar, convex, concave and saddle patches. Since grey values seen from the outside represent reflected light, we find concave functions for convex face parts. This classification facilitates many tasks in automated face analysis, demonstrated in this work by face verification on the polynomial representation. The task of face verification is to verify a face detection by analysing an image of the face.

The surface model was evaluated for segmented face approximation in 6. We are able to represent recognizable faces using a few hundred bytes rather than a few hundred kilobytes, which is very useful in visual communication applications. Moreover, the polynomial surfaces correspond to meaningful facial features. When compared to existing approximation techniques, we achieve higher

compression ratios and better recognizable faces.

The surface model was evaluated for human body analysis in 7. We elaborated curvature-based surface shape analysis for images and videos of human bodies, in order to reconstruct the human body skeleton, to detect limbs and to estimate the human pose. Human body images are represented by flat, planar, convex, concave and saddle polynomial surfaces with a variable fitting error. The low-degree polynomial surfaces correspond to meaningful human body features. We found that human body parts are often represented as nearly cylindrical surfaces, of which the axes of minimum curvature reconstruct the human body skeleton accurately. The proposed method, which provides human body skeleton reconstructions from curvature information of the surface segments, outperforms existing segmentation techniques that do not provide this information.

8.2 Future research

The methods and results presented in this PhD thesis demonstrate that the key idea of face segmentation, approximation and analysis with polynomial curves and surfaces is a well chosen strategy. However, some parts need to be further investigated to expand the possibilities of the system.

In future research, one should improve the extraction of polynomial curves and surfaces, as described in Chapter 2, in video sequences. Since the results in this work are obtained on a frame by frame basis, the polynomial curves and the boundaries of the polynomial surfaces suffer from temporal stability in video sequences. The models should be further improved by including temporal information of the shape and the location of the polynomials in the segmentation proces. A first possible solution to achieve temporal stability is to retain the supporting elemental subsets (Section 2.6.3) for corresponding surfaces in consecutive frames as long a possible. In addition, mean filtering of the polynomial coefficients over time can help to achieve stability in videos.

The methods presented in Chapter 2 and Chapter 3 are developed for greyscale images and videos. A valuable extension would be to let the methods work in the colour domain. The polynomial curve matching in Chapter 3 would then include colour information in the matching function.

The applications with polynomial curves presented in Chapter 4 and Chapter 5 should be further tested for other commercial and industrial purposes. An example is to use automated face analysis in a monitoring system for people watching advertisements on screen or watching products in a shop. The idea is to automatically find out what people are interested in, in order to adapt or personalize the presented advertisements or products.

The compression factor and the temporal stability of segmented face approximation as presented in Chapter 6 can be further improved in video sequences by

not coding the 2D boundaries of the surface segments, but by coding the projected 3D intersection curves (conics) which are found in the intersection areas of the supporting polynomials of neighouring polynomial surfaces.

The applications with polynomial surfaces, presented in Chapter 6 and Chapter 7, should be used as a helping tool to solve difficult problems that pop up in other methods. For instance in sports analysis, one problem is the segmentation of the convex hull into arms and legs. The convex hull [Slembrouck et al., 2014] is the 3D reconstruction of the shape of the human body with multiple cameras. To solve this problem, one could use the segmentation into polynomial curves and surfaces as a starting tool to find arms and legs in the 3D representation.

8.3 Summary of contributions

To summarize, the main contributions of this thesis are:

- A novel face model where the face is seen as a flexible ellipsoid mask with cutouts for the eyes, the mouth, the nose and the nostrils. The contour pixels and the image intensities of the different facial parts are represented by polynomial surfaces and curves that are convex or concave. The flexibility of the model is obtained by allowing polynomials with a variable degree and a variable approximation error.

- Novel 1-D and 2-D segmentation algorithms based on adaptive region growing and low-degree polynomial fitting to extract geometric low-level features from contour pixels and image intensities, respectively. These algorithms use a new adaptive thresholding technique with the L_∞ fitting cost as a segmentation criterion. The polynomial degree and the fitting error are automatically adapted during the region growing process. The main novelty is that the algorithms detect outliers, distinguish between strong and smooth discontinuities and find segments that are bent in a certain way, such as convex or concave segments. Adaptive refers to the use of a local neighbourhood to add pixels, while adapting the shape (or degree) of the function is based on global behaviour. In this sense there is some local flexibility, while the global behaviour is determined by a more straightforward characterization, such as being concave or convex. This work was published in [Deboeverie et al., 2010, Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- An original solution for the correspondence problem of polynomial curves approximating contours in different images. The main contribution is the introduction of intensity variations in the matching function. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009b, Deboeverie et al., 2011].

- A new way of curvature-based surface shape analysis of faces and human bodies in images. The main idea is to use the curvatures of polynomial surfaces to classify facial and human body features into flat, planar, convex, concave and saddle patches. This classification facilitates the analysis of facial and human behaviour. This work was published in [Deboeverie et al., 2013c, Deboeverie et al., 2013b].

- A novel segmented face approximation algorithm to send and store images of faces at a low bit rate, such that the faces are still recognizable and that the compression does not prevent remote face analysis. Segmented face approximation with low-degree polynomial surfaces and curves is quite natural and offers a compact and reversible way to preserve the essential characteristics of the original face image. This work was published in [Deboeverie et al., 2013c].

- A practical framework for face analysis applications, e.g. recognition and tracking of faces and facial features. We evaluated the performance of face analysis applications on a large number of representative databases and video sequences. Furthermore, we compared the proposed methods with several techniques of the state of the art. In extension, we applied our algorithms on several other objects, such as vehicles. This work was published in [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2011, Deboeverie et al., 2012].

In total, the research during this PhD resulted in three papers as first author and one paper as second author in international peer-reviewed journals [Deboeverie et al., 2013c, Deboeverie et al., 2013b, Deboeverie et al., 2013a, Bo Bo et al., 2014], of which two are published and two are submitted. Furthermore, ten conference papers as first author were published in the proceedings of international or national conferences [Deboeverie et al., 2008b, Deboeverie et al., 2008a, Deboeverie, 2008, Deboeverie et al., 2009a, Deboeverie et al., 2009b, Deboeverie et al., 2010, Deboeverie et al., 2011, Deboeverie, 2011, Deboeverie et al., 2012, Deboeverie et al., 2014] and three publications as co-author [Geelen et al., 2009, Maes et al., 2009, Eldib et al., 2014].

This work has led to important and critical contributions to the ISYSS project (Intelligent SYstems for Security and Safety), the iCocoon project (Immersive COmmunication by means of COmputer visiON). The experience gained in the projects ISYSS and iCocoon is now being used to contribute in a more responsible position to the recently started projects LittleSister (low-cost monitoring for care and retail) and SONOPA (SOcial Networks for Older adults to Promote an Active life).

Bibliography

[Adams and Bischof, 1994] Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):641–647.

[Aggarwal and Ryoo, 2012] Aggarwal, J. K. and Ryoo, M. S. (2012). Human motion analysis: A review. *ACM Computing Surveys*. To appear.

[Aharon et al., 2006] Aharon, M., Elad, M., and Bruckstein, A. (2006). The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54:4311–4322.

[Alahi et al., 2012] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 510–517.

[Alani et al., 2007] Alani, D., Averbuch, A., and Dekel, S. (2007). Image coding with geometric wavelets. *IEEE Trans. Image Processing*, 16(1):69–77.

[Albiol et al., 2008] Albiol, A., Monzo, D., Martin, A., Sastre, J., and Albiol, A. (2008). Face recognition using hog-ebgm. *Pattern Recognition Letters*, 29(10):1537–1543.

[Aravind et al., 2002] Aravind, I., Chaitanya, C., Guruprasad, M., Partha, S., and Sudhaker, S. (2002). Implementation of image segmentation and reconstruction using genetic algorithms. In *IEEE Int. Conf. on Industrial Technology*, pages 970–975.

[Arbelaez et al., 2011] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.

[ATT,] ATT. The database of faces, at&t laboratories.

[Aubrey et al., 2010] Aubrey, A., Hicks, Y., and Chambers, J. (2010). Visual voice activity detection with optical flow. *IET Image Processing*, 4(6):463–472.

[Aubrey et al., 2007] Aubrey, A., Rivet, B., Hicks, Y., Girin, L., Chambers, J., and Jutten, C. (2007). Two novel visual voice activity detectors based on appearance models and retinal filltering. In *Proc. of the 15th European Signal Processing Conference*.

[Aujol and Chan, 2006] Aujol, J. F. and Chan, T. F. (2006). Combining geometrical and textured information to perform image classification. *J. Visual Commun. Image Represent.*, 17(5):1004–1023.

[Ballard, 1981] Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.

[Barnard and Heikkila, 2008] Barnard, M. and Heikkila, J. (2008). Body part segmentation of noisy human silhouette images. In *Proc. of IEEE Int. Conf. on Multimedia and Expo*, pages 1189–1192.

[Bartlett et al., 2000] Bartlett, M. S., Donato, G. L., Movellan, J. R., Hager, J. C., Ekman, P., and Sejnowski, T. J. (2000). Image representations for facial expression coding. *Advances in Neural Information Processing Systems*, 12:886–892.

[Bartlett, 2007] Bartlett, R. (2007). *Analysing Human Movements Patterns*. Routledge.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In *Proc. of European Conf. on Computer Vision*, pages 404–417.

[Belhumeur et al., 1997] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–721.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24(24):509–522.

[Bern,] Bern. Bern university face database, university of bern.

[Besl and Jain, 1988] Besl, P. J. and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Tran. on Pattern Analysis and Machine Interlligence*, 10(2):167–192.

[Beucher, 1994] Beucher, S. (1994). Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology and Its Applications to Image Processing*, pages 69–76.

[Beucher and Meyer, 1993] Beucher, S. and Meyer, F. (1993). The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*, 34:433–481.

[Biswas, 2003] Biswas, S. (2003). Segmentation based compression for graylevel images. *Pattern Recognition*, 36(7):1501–1517.

[Bo Bo et al., 2014] Bo Bo, N., Deboeverie, F., El Dib, M., Guan, J., Xie, X., Casteñada, J. N., Van Haerenborgh, D., Slembrouck, M., Van de Velde, S., Steendam, H., Veelaert, P., Kleihorst, R., Aghajan, H., and Philips, W. (2014). Human mobility monitoring in very low-resolution visual sensor network. *MDPI special issue on Ambient Assisted Living (AAL): Sensors, Architectures and Applications.* submitted.

[Bolle et al., 1992] Bolle, R. M., Califano, A., and Kjeldsen, R. (1992). A complete and extendable approach to visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(5):534–548.

[Bosworth and Acton, 2000] Bosworth, J. and Acton, S. (2000). Segmentation-based image coding by morphological local monotonicity. In *Conf. on Signals, Systems and Computers*, pages 65–69.

[Bracewell, 1965] Bracewell, R. N. (1965). *The Fourier Transform and Its Applications*. McGraw-Hill.

[Brunelli and Poggio, 1993] Brunelli, R. and Poggio, T. (1993). Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052.

[Bryt and Elad, 2008] Bryt, O. and Elad, M. (2008). Compression of facial images using the k-svd algorithm. *J. Vis. Commun. Image R.*, 19:270–282.

[Bunke, 2000] Bunke, H. (2000). Graph matching: Theoretical foundations, algorithms, and applications. In *Vision Interface*, pages 82–88.

[Cabrera et al., 2012] Cabrera, R. R., Tuytelaars, T., and Gool, L. V. (2012). Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application. *Computer Vision and Image Understanding*, 116(6):742–753.

[Cadzow, 2002] Cadzow, J. A. (2002). Minimum l_1, l_2 and l_∞ norm approximate solutions to an overdetermined system of linear equations. *Digital Signal Processing*, 12(4):524–560.

[Candès and Donoho, 1999] Candès, E. J. and Donoho, D. L. (1999). Curvelets: A surprisingly effective nonadaptive representation for objects with edges. In *Curves and Surfaces*, pages 1–10.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679–698.

[Cevikalp et al., 2005] Cevikalp, H., Neamtu, M., Wilkes, M., and Barkana, A. (2005). Discriminative common vectors for face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(1):4–13.

[Cha, 2007] Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *Int. Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307.

[Cha and Srihari, 2002] Cha, S.-H. and Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370.

[Chen et al., 2005] Chen, J., Pappas, T. N., Mojsilovic, A., and Rogowitz, B. E. (2005). Adaptive perceptual color-texture image segmentation. *IEEE Trans. on Image Processing*, 14(10):1524–1536.

[Christopoulos et al., 1997] Christopoulos, C. A., Philips, W., Skodras, A. N., and Cornelis, J. (1997). Segmented image coding: Techniques and experimental results. *Signal Processing: Image Communication*, 11:63–80.

[Cohen-Steiner et al., 2004] Cohen-Steiner, D., Alliez, P., and Desbrun, M. (2004). Variational shape approximation. *ACM Trans. on Graphics (SIGGRAPH)*, 23(3):905–914.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

[Cootes et al., 2001] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

[Cootes et al., 1992] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1992). Proc. of british machine vision conference. In *Training Models of Shape from Sets of Examples*, pages 9–18.

[Cootes et al., 2002] Cootes, T. F., Wheeler, G. V., Walker, K. N., and Taylor, C. J. (2002). View-based active appearance models. *Image and vision computing*, 20(9):657–664.

[Couprie et al., 2011] Couprie, C., Grady, L., Najman, L., and Talbot, H. (2011). Power watershed: A unifying graph-based optimization framework. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(7):1384–1399.

[Cour and Shi, 2007] Cour, T. and Shi, J. (2007). Recognizing objects by piecing together the segmentation puzzle. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8.

[Cox et al., 1996] Cox, J., Ghosn, J., and Yianilos, P. N. (1996). Proc. of computer vision and pattern recognition. In *Feature-Based Face Recognition Using Mixture-Distance*, pages 209–216.

[Cristinacce and Cootes, 2008] Cristinacce, D. and Cootes, T. F. (2008). Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 886–893.

[Daniyal et al., 2010] Daniyal, F., Taj, M., and Cavallaro, A. (2010). Content and task-based view selection from multiple video streams. *Multimedia Tools and Applications*, 46:235–258.

[Deboeverie, 2008] Deboeverie, F. (2008). Shape matching with geometric primitives. In *9th UGent FirW Doctoraatssymposium*, pages 156–157. Universiteit Gent. Faculteit Ingenieurswetenschappen.

[Deboeverie, 2011] Deboeverie, F. (2011). A uniform approach for face segmentation and coding with adaptive region growing. In *12th UGent FEA Doctoraatssymposium*, page 48. Universiteit Gent. Faculteit Ingenieurswetenschappen en Architectuur.

[Deboeverie et al., 2014] Deboeverie, F., Allebosch, G., Van Haerenborgh, D., Veelaert, P., and Philips, W. (2014). Edge-based foreground detection with higher order derivative local binary patterns for low-resolution video processing. In *Proc. of Int. Conf. on Computer Vision Theory and Applications*, pages 339–346.

[Deboeverie et al., 2009a] Deboeverie, F., Maes, F., Veelaert, P., and Philips, W. (2009a). Linked geometric features for modeling the fluid flow in developing embryonic vertebrate hearts. In *Proc. of Int. Conf. on Image Processing*, pages 2473–2476. IEEE.

[Deboeverie et al., 2009b] Deboeverie, F., Teelen, K., Veelaert, P., and Philips, W. (2009b). Vehicle tracking using geometric features. In *Proc. of Advanced Concepts for Intelligent Vision Systems*, volume 5807, pages 506–515. Springer.

[Deboeverie et al., 2010] Deboeverie, F., Teelen, K., Veelaert, P., and Philips, W. (2010). Adaptive constructive polynomial fitting. In *Proc. of Advanced Concepts for Intelligent Vision Systems*, volume 6474, pages 173–184. Springer.

[Deboeverie et al., 2008a] Deboeverie, F., Veelaert, P., and Philips, W. (2008a). Parabola-based face recognition and tracking. In *Proc. of Annual Workshop on Semiconductor Advances for Future Electronics and Sensors*, pages 308–313. STW Technology Foundation.

[Deboeverie et al., 2011] Deboeverie, F., Veelaert, P., and Philips, W. (2011). Face analysis using curve edge maps. In *Proc. of Int. Conf. on Image Analysis and Processing*, volume 6979, pages 109–118. Springer.

[Deboeverie et al., 2012] Deboeverie, F., Veelaert, P., and Philips, W. (2012). Best view selection with geometric feature based face recognition. In *Proc. of Int. Conf. on Image Processing*, pages 1461–1464.

[Deboeverie et al., 2013a] Deboeverie, F., Veelaert, P., and Philips, W. (2013a). Human body parts segmentation in physiotherapy with adaptive curvature-based region growing. *International Journal of Computer Vision*. submitted.

[Deboeverie et al., 2013b] Deboeverie, F., Veelaert, P., and Philips, W. (2013b). Image segmentation with adaptive region growing based on a polynomial surface model. *Journal of Electronic Imaging*, 22(4). DOI:10.1117/1.JEI.22.4.043004.

[Deboeverie et al., 2013c] Deboeverie, F., Veelaert, P., and Philips, W. (2013c). Segmented face approximation with adaptive region growing based on low-degree polynomial fitting. *Signal, Image and Video Processing*. DOI:10.1007/s11760-013-0441-6.

[Deboeverie et al., 2008b] Deboeverie, F., Veelaert, P., Teelen, K., and Philips, W. (2008b). Face recognition using parabola edge map. In *Proc. of Advanced Concepts for Intelligent Vision Systems*, volume 5259, pages 994–1005. Springer.

[Deng and Manjunath, 2001] Deng, Y. and Manjunath, B. S. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):800–810.

[Déniz et al., 2011] Déniz, O., Bueno, G., Salido, J., and De la Torre, F. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603.

[Deschaud and Goulette, 2006] Deschaud, J. E. and Goulette, F. (2006). A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In *Proceedings of the Fifth Int. Symposium on 3D Data Processing, Visualization and Transmission*, pages 248–253.

[Ding and Martinez, 2010] Ding, L. and Martinez, A. M. (2010). Features versus context: An approach for precise and detailed detection and delineation of faces and facial features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(11):2022–2038.

[Do and Vetterli, 2003] Do, M. N. and Vetterli, M. (2003). *Contourlets: Beyond Wavelets*. J. Stoeckler and G. V. Welland, Eds. San Diego, CA: Academic.

[Donoho, 1999] Donoho, D. L. (1999). Wedgelets: Nearly minimax estimation of edges. *Ann. Statist.*, 27(3):859–897.

[Donoho and Huo, 2001] Donoho, D. L. and Huo, X. (2001). Beamlets and multiscale image analysis. Technical report, Department of Statistics, Stanford Univ., Stanford, CA.

[Dreuw et al., 2009] Dreuw, P., Steingrube, P., Hanselmann, H., and Ney, H. (2009). Proc. of british machine vision conference. In *SURF Face: Face Recognition under Viewpoint Consistency Constraints*, pages 1–11.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Comm. of the ACM*, 15:11–15.

[Eden et al., 2005] Eden, E., Waisman, D., Rudzsky, M., Bitterman, H., Brod, V., and Rivlin, E. (2005). An automated method for analysis of flow characteristics of circulating particles from in vivo video microscopy. *IEEE Trans. Med. Imaging*, 24(8):1011–1024.

[Eden and Kocher, 1985] Eden, M. and Kocher, M. (1985). On the performance of a contour coding algorithm in the context of image coding part 1: contour segment coding. *Signal Processing*, 8:381–386.

[Elad et al., 2007] Elad, M., Goldenberg, R., and Kimmel, R. (2007). Low bit-rate compression of facial images. *IEEE Trans. on Image Processing*, 16(9):2379–2383.

[Eldib et al., 2014] Eldib, M., Bo Bo, N., Deboeverie, F., Castañeda, J. N., Guan, J., Van de Velde, S., Steendam, H., Aghajan, H., and Philips, W. (2014). A low resolution multi-camera system for person tracking. In *Int. Conf. on Image Processing*. accepted.

[Fan et al., 1989] Fan, T.-J., Medioni, G., and Nevatia, R. (1989). Recognizing 3-d objects using surface descriptions. *IEEE Trans. on Patterns Analysis and Machine Intelligence*, 11(11):1140–1157.

[Feris et al., 2007] Feris, R., Tian, Y.-L., and Hampapur, A. (2007). Capturing people in surveillance video. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8.

[Ferrari et al., 2008] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(1):36–51.

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

[Fu and Mui, 1981] Fu, K. S. and Mui, J. K. (1981). A survey on imagesegmentation. *Pattern Recognition*, 13(1):3–16.

[Gao and Leung, 2002a] Gao, Y. and Leung, M. K. (2002a). Face recognition using line edge map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):764–779.

[Gao and Leung, 2002b] Gao, Y. and Leung, M. K. (2002b). Face recognition using line edge map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):764–779.

[Geelen et al., 2009] Geelen, B., Deboeverie, F., and Veelaert, P. (2009). Implementation of canny edge detection on the wica smartcam architecture. In *Proc. of Int. Conf. on Distributed Smart Cameras*, pages 485–492. IEEE.

[Geng and Jiang, 2013] Geng, C. and Jiang, X. (2013). Fully automatic face recognition framework based on local and global features. *Machine Vision and Applications*, pages 1–13.

[Gerek and Çinar, 2004] Gerek, O. N. and Çinar, H. (2004). Segmentation based coding of human face images for retrieval. *Signal Processing*, 84:1041–1047.

[Gersho and Gray, 1995] Gersho, A. and Gray, R. (1995). *Vector Quantization and Signal Compression*. Kluwer Academic Publishers.

[Gilge et al., 1989] Gilge, M., Engelhardt, T., and Mehlan, R. (1989). Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing: Image Communication*, 1(2):153–180.

[Gonzalez and Woods, 2001] Gonzalez, R. C. and Woods, R. E. (2001). *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Boston, MA, USA, 2nd edition.

[Graver, 1991] Graver, J. E. (1991). Rigidity matroids. *SIAM J. Discrete Math.*, 4:355–368.

[GTFD,] GTFD. Georgia tech face database, georgia institute of technology.

[Haralick and Shapiro, 1985] Haralick, R. and Shapiro, L. G. (1985). Survey: Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proc. of Alvey Vision Conference*, pages 147–151.

[He et al., 2005] He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H. J. (2005). Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):328–340.

[Hou and Pang, 2011] Hou, Y.-L. and Pang, G. K. H. (2011). People counting and human detection in a challenging situation. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(1):24–33.

[Hough, 1962] Hough, P. V. C. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.

[Hsieh et al., 2010] Hsieh, J.-W., Chuang, C.-H., Chen, S.-Y., Chen, C.-C., and Fan, K.-C. (2010). Segmentation of human body parts using deformable triangulation. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(3):596–610.

[Hu et al., 1996] Hu, J., Wang, R., and Wang, Y. (1996). Compression of personal identification pictures using vector quantization with facial feature correction. *Optical Engineering*, 35:198–203.

[Hu et al., 2006] Hu, Z., Lin, X., and Yan, H. (2006). Torso detection in static images. In *Proc. of Int. Conf. on Signal Processing*.

[Hu et al., 2009] Hu, Z., Wang, G., Lin, X., and Yan, H. (2009). Recovery of upper body poses in static images based on joints detection. *Pattern Recognition Letters*, 30:503–512.

[Huffman, 1952] Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. *Proc. IRE*, 40(10):1098–1101.

[Jackway, 1996] Jackway, P. (1996). Gradient watersheds in morphological scalespace. *IEEE Trans. on Image Processing*, 15(6):913–921.

[Jagannathan and Miller, 2007] Jagannathan, A. and Miller, E. L. (2007). Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Tran. on Pattern Analysis and Machine Interlligence*, 29(12):2195–2204.

[Jeannin and Bober, 1999] Jeannin, S. and Bober, M. (1999). Description of core experiments for mpeg-7 motion/shape. Technical report, Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99/N2690.

[Jelaca et al., 2013] Jelaca, V., Pizurica, A., Castaneda, J. N., Velazquez, A. F., and Philips, W. (2013). Vehicle matching in smart camera networks using image projection profiles at multiple instances. *Image and Vision Computing*, 31(9):673–685.

[Jeong et al., 2006] Jeong, J. H., Sugii, Y., Minamiyama, M., and Okamoto, K. (2006). Measurement of rbc deformation and velocity in capillaries in vivo. *Microvasc. Res.*, 71(3):212–217.

[Jesorsky et al., 2001] Jesorsky, O., Kirchberg, K. J., and Frischholz, R. W. (2001). Robust face detection using the hausdorff distance. In *Proc. of Int. Conf. on Audio- and Videobased Biometric Person Authentication*, pages 90–95.

[Jiang et al., 2008] Jiang, H., Fels, S., and Little, J. (2008). Optimizing multiple object tracking and best view video synthesis. *IEEE Trans. on Multimedia*, 10(6):997–1012.

[Juang et al., 2009] Juang, C.-F., Chang, C.-M., Wu, J.-R., and Lee, D. (2009). Computer vision-based human body segmentation and posture estimation. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(1):119–133.

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

[Kanga et al., 2012] Kanga, C.-C., Wanga, W.-J., and Kangb, C.-H. (2012). Image segmentation with complicated background by using seeded region growing. *Int. J. Electron. Commun.*

[Kapur et al., 1985] Kapur, J. N., Sahoo, P. K., and Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285.

[Kassim et al., 2009] Kassim, A. A., Lee, W. S., and Zonoobi, D. (2009). Hierarchical segmentation-based image coding using hybrid quad-binary trees. *IEEE Trans. on Image Processing*, 6:1284–1291.

[Kelley and Weiss, 1979] Kelley, P. J. and Weiss, M. L. (1979). *Geometry and Convexity: A Study in Mathematical Methods*. Wiley.

[Kelly et al., 2009] Kelly, P., Conaire, C. O., Kim, C., and O'Connor, N. E. (2009). Automatic camera selection for activity monitoring in a multi-camera system for tennis. In *Proc. Int. Conf. on Distributed Smart Cameras*.

[Kim et al., 2005] Kim, J., Choi, J., Yi, J., and Turk, M. (2005). Effective representation using ica for face recognition robust to local distortion and partial occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1997–1921.

[Kim and Kittler, 2005] Kim, T.-K. and Kittler, J. (2005). Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3):1–10.

[Kocher and Leonardi, 1986] Kocher, M. and Leonardi, R. (1986). Adaptive region growing technique using polynomial functions for image approximation. *Signal Processing*, 11:47–60.

[Kunt et al., 1987] Kunt, M., Benard, M., and Leonardi, R. (1987). Recent results in high-compression image coding. *IEEE Trans. on Circuits and Systems*, 34(11):1306–1336.

[Lades et al., 1993] Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, M. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers*, 42(3):300–311.

[Lambert, 1760] Lambert, J. H. (1760). *Photometria Sive de Mensura de Gratibus Luminis*. Colorum et Umbrae. Augsberg, Germany: Eberhard Klett.

[Lanitis et al., 1995] Lanitis, A., Taylor, C. J., and Cootes, T. F. (1995). A unified approach to coding and interpreting face images. In *IEEE Int. Conf. on Computer Vision*, pages 368–373.

[Lanitis et al., 1997] Lanitis, A., Taylor, C. J., and Cootes, T. F. (1997). Automatic interpretation and coding of face images using flexible models. *IEEE Trans. on Pattern and Machine Intelligence*, 19(7):743–756.

[Lavoué et al., 2005] Lavoué, G., Dupont, F., and Baskurt, A. (2005). A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10):975–987.

[Lee and Cohen, 2006] Lee, M. W. and Cohen, I. (2006). A model-based approach for estimating human 3d poses in static images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(6):905–916.

[Leutenegger et al., 2011] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *IEEE Int. Conf. on Computer Vision*, pages 2548–2555.

[Li et al., 2005] Li, C., Xu, C., Gui, C., and Fox, M. D. (2005). Level set evolution without re-initialization: A new variational formulation. In *Proc. of Computer Vision and Pattern Recognition*, pages 430–436.

[Li et al., 1993] Li, H., Roivainen, P., and Forchheimer, R. (1993). 3-d motion estimation in model-based facial image coding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):545–555.

[Li et al., 2011] Li, S., Lu, H.-C., Ruan, X., and Chen, Y.-W. (2011). Human body segmentation based on deformable models and two-scale superpixel. *Pattern Analysis and Applications*, pages 1–15.

[Li and Bhanu, 2009] Li, Y. and Bhanu, B. (2009). Task-oriented camera assignment in a video network. In *Proc. of Int. Conf. on Image Processing*, pages 3473–3476.

[Liang et al., 2008] Liang, L., Xiao, R., Wen, F., and Sun, J. (2008). Face alignment via component-based discriminative search. In *Proc. of European Conference on Computer Vision*, pages 72–85.

[Liang et al., 2009] Liang, Y.-M., Shih, S.-W., Shih, C.-C., Liao, H.-Y. M., and Lin, C.-C. (2009). Learning atomic human actions using variable-length markov models. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(1):268–280.

[Lim et al., 1990] Lim, Y. S., Cho, T. I., and Park, K. (1990). Range image segmentation based on 2d quadratic function approximation. *Pattern Recognition Letters*, 11(10):699–708.

[Liu et al., 2011a] Liu, Q., Li, H., and Ngan, K. N. (2011a). Automatic body segmentation with graph cut and self-adaptive initialization level set (sails). *J. Vis. Commun. Image R.*, 22:367–377.

[Liu, 2007] Liu, X. (2007). Generic face alignment using boosted appearance model. In *Proc. of Computer Vision and Pattern Recognition*, pages 1079–1088.

[Liu, 2009] Liu, X. (2009). Discriminative face alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(11):1941–1954.

[Liu et al., 2011b] Liu, Z., Shen, L., and Zhang, Z. (2011b). Unsupervised image segmentation based on analysis of binary partition tree for salient object extraction. *Signal Processing*, 91(2):290–299.

[Loncaric, 1998] Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001.

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

[Lyons and Akamatsu, 1998] Lyons, M. and Akamatsu, S. (1998). Coding facial expressions with gabor wavelets. In *Proc., Third IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 200–205.

[Maes et al., 2009] Maes, F., Deboeverie, F., Van Ransbeeck, P., and Verdonck, P. (2009). Tools to understand the pumping mechanism of embryonic hearts. In *Proc. of Int. Conf. on Computational Bioengineering*, pages 86–86.

[Makrogiannis et al., 2005] Makrogiannis, S., Economou, G., and Fotopoulos, S. (2005). A region dissimilarity relation that combines feature-space and spatial information for color image segmentation. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 358(1):44–53.

[Martin et al., 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 2, pages 416–423.

[Martinez, 2002] Martinez, A. M. (2002). Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):748–763.

[Martínez, 2003] Martínez, A. M. (2003). Matching expression variant faces. *Vision Research*, 43:1047–1060.

[Martinez and Benavente, 1998] Martinez, A. M. and Benavente, R. (1998). The ar face database. Technical report, CVC Technical Report #24.

[Matas and Chum, 2004] Matas, J. and Chum, O. (2004). Randomized ransac with td,d test. *Image and Vision Computing*, 22(10):837–842.

[Maxwell and Shafer, 2000] Maxwell, B. A. and Shafer, S. A. (2000). Segmentation and interpretation of multicolored objects with highlights. *Computer Vision and Image Understanding*, 77:1–24.

[Mian et al., 2007] Mian, A. S., Bennamoun, M., and Owens, R. (2007). An efficient multimodel 2d-3d hybrid approach to automatic face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(11):1927–1943.

[Mignotte, 2008] Mignotte, M. (2008). Segmentation by fusion of histogram-based k-means clusters in different color spaces. *IEEE Trans. on Image Processing*, 17(5):780–787.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. J. V. (2005). A comparison of affine region detectors. *Int. Journal of Computer Vision*, 65(1-2):43–72.

[Moghaddam and Pentland, 1995] Moghaddam, B. and Pentland, A. (1995). An automatic system for model-based coding of faces. In *IEEE Data Compression Conference*, pages 362–370.

[Moravec, 1977] Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, page 584.

[Morel and Yu, 2009] Morel, J. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *Journal on Imaging Sciences*, 2(2):438–469.

[Mori and Malik, 2002] Mori, G. and Malik, J. (2002). Estimating human body configurations using shape context matching. In *Proc. of European Conf. on Computer Vision*, pages 666–680.

[Mori and Malik, 2006] Mori, G. and Malik, J. (2006). Recovering 3d human body configurations using shape contexts. *IEEE Trans. Pattern Analysis Machine Intelligence*, 28(7):1052–1062.

[Mori et al., 2004] Mori, G., Ren, X., Efros, A. A., and Malik, J. (2004). Recovering human body configuration: Combining segmentation and recognition. In *Proc. of Computer Vision and Pattern Recognition*, pages 326–333.

[Muñoz et al., 2003] Muñoz, X., Freixenet, J., Cufí, X., and Martí, J. (2003). Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters*, 24(1-3):375–392.

[Murphy-Chutorian and Trivedi, 2009] Murphy-Chutorian, E. and Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4):607–626.

[Nevatia, 1986] Nevatia, R. (1986). Image segmentation. *Handbook of Pattern Recognition and Image Processing*, 86:215–231.

[Nixon and Aguado, 2012] Nixon, M. S. and Aguado, A. S. (2012). *Feature Extraction and Image Processing for Computer Vision*. Academic Press.

[Ojala et al., 2000] Ojala, T., Pietikainen, M., and Maenpaa, T. (2000). Gray scale and rotation invariant texture classification with local binary patterns. In *Proc. of European Conf. on Computer Vision*, pages 404–420.

[Otsu, 1979] Otsu, N. (1979). A threshold selection method from grey-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66.

[Pal and Pal, 1993] Pal, N. R. and Pal, S. K. (1993). A review of image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294.

[Pantofaru and Hebert, 2005] Pantofaru, C. and Hebert, M. (2005). A comparison of image segmentation algorithms. Technical Report CMU-RI-TR-05-40, Robotics Institute, Pittsburgh, PA.

[Pappas, 1992] Pappas, T. N. (1992). An adaptive clustering algorithm for image segmentation. *IEEE Trans. on Signal Processing*, 40(4):901–914.

[Park et al., 1999] Park, J.-S., Oh, H.-S., Chang, D.-H., and Lee, E.-T. (1999). Human posture recognition using curve segments for image retrieval. In *Proc. of SPIE*.

[Park and Aggarwal, 2004] Park, S. and Aggarwal, J. K. (2004). Semantic-level understanding of human actions and interactions using event hierarchy. In *Proc. of Computer Vision and Pattern Recognition Workshop*, page 12.

[Pennec and Mallat, 2005] Pennec, E. L. and Mallat, S. (2005). Sparse geometric image representations with bandelets. *IEEE Trans. Image Processing*, 14(4):423–438.

[Petitjean, 2002] Petitjean, S. (2002). A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 34(2):211–262.

[Philips, 1996] Philips, W. (1996). Fast coding of arbitrarily shaped image segments using weakly separable bases. *Optical Engineering, Special section on Visual Communications and Image Processing*, 35:177–186.

[Pluim et al., 2001] Pluim, J. P. W., Maintz, J. B. A., and Viergever, M. A. (2001). Mutual information matching in multiresolution contexts. *Image and Vision Computing*, 19(1-2):45–52.

[Pohle and Toennies, 2001] Pohle, R. and Toennies, K. D. (2001). Proc. of spie medical imaging. In *Segmentation of medical images using adaptive region growing*, pages 1337–1347.

[Porikli and Tuzel, 2003] Porikli, F. and Tuzel, O. (2003). Human bodytracking byadaptive background models and mean-shift analysis. In *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*.

[Pratt, 1991] Pratt, W. K. (1991). *Digital Image Processing, 2nd ed.* Wiley.

[Qin and Clausi, 2010] Qin, A. K. and Clausi, D. A. (2010). Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Trans. on Image Processing*, 19(8):2157–2170.

[Rabbani et al., 2006] Rabbani, T., van den Heuvel, F. A., and Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint. In *ISPRS Commission V Symposium Image Engineering and Vision Metrology*, pages 248–253.

[Radha et al., 1996] Radha, H., Vetterli, M., and Leonardi, R. (1996). Image compression using binary space partitioning trees. *IEEE Trans. Image Processing*, 5(12):1610–1624.

[Rangarajan et al., 1999] Rangarajan, A., Chui, H., and Duncan, J. S. (1999). Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440.

[Reid et al., 1997] Reid, M. M., Millar, R. J., and Black, N. D. (1997). Second-generation image coding: An overview. *ACM Computing Surveys*, 29(1):3–29.

[Roberts and Marshall, 1998] Roberts, D. R. and Marshall, A. D. (1998). Proc. of british machine vision conference. In *Viewpoint selection for complete surface coverage of three dimensional objects*, pages 740–750.

[Rosten and Drummond, 2005] Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *IEEE Int. Conf. on Computer Vision*, pages 1508–15011.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *European Conf. on Computer Vision*, pages 430–443.

[Roth and Levine, 1993] Roth, G. and Levine, M. (1993). Extracting geometric primitives. *CVGIP:Image Understanding*, 58(1):1–22.

[Ruiz-del Solar et al., 2009] Ruiz-del Solar, J., Verschae, R., and Correa, M. (2009). Recognition of faces in unconstrained environments: A comparative study. *EURASIP Journal on Advances in Signal Processing*, pages 1–20.

[Ruppertsberg et al., 1998] Ruppertsberg, A. I., Vetter, T., and Bülthoff, H. H. (1998). A face specific similarity measure for image coding and synthesis. *Investigative Ophthalmology & Visual Science*, 39(4):173.

[Sakalli and Yan, 1998] Sakalli, M. and Yan, H. (1998). Feature-based compression of human face images. *Optical Engineering*, 37:1520–1529.

[Salembier et al., 1996] Salembier, P., Brigger, P., Casas, J., and Pardas, M. (1996). Morphological operators for image and video compression. *IEEE Trans. on Image Processing*, 5(6):881–898.

[Samet, 1984] Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Comput. Surv.*, 16(2):187–260.

[Sbalzarini and Koutmoutsakos, 2005] Sbalzarini, L. F. and Koutmoutsakos, P. (2005). Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.*, 151(2):182–195.

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

[Shan et al., 2008] Shan, Y., Sawhney, H. S., and Kumar, R. T. (2008). Unsupervised learning of discriminative edge measures for vehicle matching between nonoverlapping cameras. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(4):894–901.

[Shao et al., 2012] Shao, L., Ji, L., Liu, Y., and Zhang, J. (2012). Human action segmentation and recognition via motion and shape analysis. *Pattern Recognition Letters*, 33:438–445.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Shih and Cheng, 2005] Shih, F. Y. and Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, 23(10):877–886.

[Shukla et al., 2005] Shukla, R., Dragotti, P. L., Do, M. N., and Vetterli, M. (2005). Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. *IEEE Trans. on Image Processing*, 14(3):343–359.

[Simari and Singh, 2005] Simari, P. and Singh, K. (2005). Proc. of graphics interface. In *Extraction and remeshing of ellipsoidal representations from mesh data*, pages 161–168.

[Slembrouck et al., 2014] Slembrouck, M., Van Cauwelaert, D., Van Hamme, D., Van Haerenborgh, D., Van Hese, P., Veelaert, P., and Philips, W. (2014). Self-learning voxel-based multi-camera occlusion maps for 3d reconstruction. In *Proc. of Int. Conf. on Computer Vision Theory and Applications*.

[Smith and Brady, 1997] Smith, S. and Brady, J. (1997). Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

[Somasundarama and Palaniappan, 2011] Somasundarama, K. and Palaniappan, N. (2011). Adaptive low bit rate facial feature enhanced residual image coding method using spiht for compressing personal id images. *Int. Journal of Electronics and Communications*, 65:589–594.

[Srinivasan and Shi, 2007] Srinivasan, P. and Shi, J. (2007). Bottom-up recognition and parsing of the human body. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8.

[Stirling,] Stirling. Psychological image collection at stirling.

[Stoer and Witzgall, 1970] Stoer, J. and Witzgall, C. (1970). *Convexity and Optimization in Finite Dimensions*. Springer-Verlag.

[Stromberg, 1993] Stromberg, A. (1993). Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression. *SIAM Journal on Scientific Computing*, 14(6):1289–1299.

[Sullivan and Baker, 1994] Sullivan, G. J. and Baker, R. L. (1994). Efficient quadtree coding of images and video. *IEEE Trans. Image Process.*, 3(3):327–331.

[Takács, 1998] Takács, B. (1998). Comparing face images using the modified hausdorff distance. *Pattern Recognition*, 31(12):1873–1881.

[Tan and Triggs, 2007] Tan, X. and Triggs, B. (2007). Analysis and modeling of faces and gestures. In *Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition*, pages 235–249.

[Tao et al., 2007] Tao, W., Jin, H., and Zhang, Y. (2007). Color image segmentation based on mean shift and normalized cuts. *IEEE Trans. on Systems, Man, and Cybernetics*, 37(5):1382–1389.

[Tarabalka et al., 2010] Tarabalka, Y., Chanussot, J., and Benediktsson, J. A. (2010). Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7):2367–2379.

[Taubin, 1991] Taubin, G. (1991). Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138.

[Taubman and Marcellin, 2002] Taubman, D. and Marcellin, M. (2002). *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer.

[Teelen, 2010] Teelen, K. (2010). *Geometric uncertainty models for correspondence problems in digital image processing*. PhD thesis.

[Tessens et al., 2008] Tessens, L., Morbee, M., Lee, H., Philips, W., and Aghajan, H. (2008). Principal view determination for camera selection in distributed smart camera networks. In *Proc. Int. Conf. on Distributed Smart Cameras*.

[Tou and Gonzalez, 1974] Tou, J. T. and Gonzalez, R. C. (1974). *Pattern Recognition Principles*. Reading MA: Addison-Wesley.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

[Tuytelaars and Mikolajczyk, 2008] Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280.

[Vázquez et al., 2003] Vázquez, P.-P., Feixas, M., Sbert, M., and Heidrich, W. (2003). Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum*, 22(4):689–700.

[Veelaert, 1993] Veelaert, P. (1993). On the flatness of digital hyperplanes. *Journal of Mathematical Imaging and Vision*, 3(2):205–221.

[Veelaert, 1994] Veelaert, P. (1994). Recognition of digital algebraic surfaces by large collecions of inequalities. In *Proc. of the SPIE Conference on Vision Geometry III*, volume 2356, pages 2–11.

[Veelaert, 1997] Veelaert, P. (1997). Constructive fitting and extraction of geometric primitives. *CVGIP: Graphical Models and Image Processing*, 59(4):233–251.

[Veelaert, 2012] Veelaert, P. (2012). Separability and tight enclosure of point sets, digital geometry algorithms. *Theoretical Foundations and Applications to Computational Imaging*, 2:215–243.

[Veelaert and Teelen, 2006] Veelaert, P. and Teelen, K. (2006). Fast polynomial segmentation of digitized curves. In *Proc. of Discrete Geometry for Computer Imagery*, volume 4245, pages 482–493.

[Vieira and Shimada, 2005] Vieira, M. and Shimada, K. (2005). Surface mesh segmentation and smooth surface extraction through region growing. *Computer-Aided Geometric Design*, 22(8):771–792.

[Vila-Forcén et al., 2006] Vila-Forcén, J. E., Voloshynovskiy, S., Koval, O., and Pun, T. (2006). Facial image compression based on structured codebooks in overcomplete domain. *EURASIP Journal on Applied Signal Processing*, 2006:69042:1–11.

[Vincent and Soille, 1991] Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. of Computer Vision and Pattern Recognition*, pages 511–518.

[Viola and Wells, 1997] Viola, P. and Wells, W. M. (1997). Alignment by maximization of mutual information. *Int. Journal of Computer Vision*, 24(2):137–154.

[W. Xiaoyu, 2009] W. Xiaoyu, H. Tony, Y. S. (2009). An hog-lbp human detector with partial occlusion handling. In *Proc. of Int. Conf. on Computer Vision*, pages 32–39.

[Wagemans et al., 2010] Wagemans, J., Doorn, A. J. V., and Koenderink, J. (2010). The shading cue in context. *i-Perception*, 1(3):159–178.

[Wang et al., 2003] Wang, G., Houkes, Z., Ji, G., Zheng, B., and Li, X. (2003). An estimation-based approach for range image segmentation: on the reliability of primitive extraction. *Pattern Recognition*, 36(1):157–169.

[Wang and Yagi, 2008] Wang, J. and Yagi, Y. (2008). Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Trans. On Image Processing*, 17(2):235–240.

[Wang and Yu, 2011] Wang, J. and Yu, Z. (2011). Geometric decomposition of 3d surface meshes using morse theory and region growing. *Int J Adv Manuf Technol*, 56:1091–1103.

[Wang et al., 2012] Wang, L., Cao, J., and Han, C. (2012). Multidimensional particle swarm optimization-based unsupervised planar segmentation algorithm of unorganized point clouds. *Pattern Recognition*.

[Wang and Siskind, 2003] Wang, S. and Siskind, J. M. (2003). Image segmentation with ratio cut. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):675–690.

[Wang, 2002] Wang, W. (2002). Modelling and processing with quadric surfaces. *Handbook of Computer Aided Geometric Design*, pages 777–795.

[Watson, 1998] Watson, G. (1998). Choice of norms for data fitting and function approximation. *Acta Numerica*, 7:337–377.

[Watson, 2000] Watson, G. A. (2000). Approximation in normed linear spaces. *Journal of Computational and Applied Mathematics*, 121:1–36.

[Willett and Nowak, 2003] Willett, R. and Nowak, R. (2003). Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Trans. Medical Imaging*, 22(3):332–350.

[Wiskott and von der Malsburg, 1996] Wiskott, L. and von der Malsburg, C. (1996). Recognizing faces by dynamic link matching. *Special issue of NeuroImage*, 4(3):S14–S18.

[Wolf et al., 2008] Wolf, L., Hassner, T., and Taigman, Y. (2008). Proc. of european conf. on computer vision. In *Descriptor based Methods in the Wild*.

[Wright and Hua, 2009] Wright, J. and Hua, G. (2009). Proc. of computer vision and pattern recognition. In *Implicit Elastic Matching with Random Projections for Pose-Variant Face Recognition*, pages 1502–1509.

[Wu and Aghajan, 2007] Wu, C. and Aghajan, H. (2007). Model-based image segmentation for multi-view human gesture analysis. In *Proc. of Advanced Concepts for Intelligent Vision Systems*, pages 310–321.

[Wu and Kobbelt, 2005] Wu, J. and Kobbelt, L. (2005). Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum (EUROGRAPHICS)*, 24(3):277–284.

[Xiong and Debrunner, 2004] Xiong, T. and Debrunner, C. (2004). Stochastic car tracking with line- and colour-based features. *IEEE Trans. On Intelligent Transportation Systems*, 5(4):999–1003.

[Xu et al., 2009] Xu, C., Liu, J., and Tang, X. (2009). 2d shape matching by contour flexibility. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(1):180–186.

[Xu et al., 2008] Xu, Z., Chen, H., Zhu, S.-C., and Luo, J. (2008). A hierarchical compositional model for face representation and sketching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6):955–969.

[Yale,] Yale. Yale university face database, university of yale.

[Yan et al., 2012] Yan, D.-M., Wang, W., L., Y., and Yang, Z. (2012). Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design*. doi:10.1016/j.cad.2012.04.005.

[Yang et al., 2008] Yang, A. Y., Wright, J., Ma, Y., and Sastry, S. S. (2008). Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225.

[Yemez et al., 1997] Yemez, Y., Sankur, B., and Anarim, E. (1997). An object-oriented video codec based on region growing motion segmentation. In *Proc. of Int. Conf. on Image Processing*, pages 444–447.

[Yoo et al., 2002] Yoo, H.-W., Junga, S.-H., Janga, D.-S., and Nab, Y.-K. (2002). Extraction of major object features using vq clustering for content-based image retrieval. *Pattern Recognition*, 35:1115–1126.

[Yuille et al., 1992] Yuille, A. L., Hallinan, P. W., and Cohen, D. S. (1992). Feature extraction from faces using deformable templates. *Int. Journal of Computer Vision*, 8(2):99–111.

[Zhang et al., 2010] Zhang, B., Gao, Y., Zhao, S., and Liu, J. (2010). Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Trans. on Image Processing*, 19(2):533–544.

[Zhao et al., 2003] Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P. J. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458.

[Zou et al., 2007] Zou, J., Ji, Q., and Nagy, G. (2007). A comparative study of local matching approach for face recognition. *IEEE Trans. on Image Processing*, 16(10):2617–2628.

[Zucker, 1976] Zucker, S. (1976). Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399.