A Collaborative Approach to Image Segmentation and Behavior Recognition from Image Sequences

THÈSE N^{O} 4239 (2008)

PRÉSENTÉE LE 28 NOVEMBRE 2008
À LA FACULTE SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRAITEMENT DES SIGNAUX 5
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Laura Ioana GUI

acceptée sur proposition du jury:

Prof. A. Billard, présidente du jury
Prof. J.-Ph. Thiran, directeur de thèse
Dr X. Bresson, rapporteur
Prof. N. Paragios, rapporteur
Prof. P. Vandergheynst, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2008

Acknowledgments

At the end of this challenging journey, I would like to express my gratitude to all the people that helped me reach this moment.

First of all, I would like to thank my thesis advisor, Prof. Jean-Philippe Thiran, for his guidance and enthusiasm with respect to my work, and especially for his warm smile and for all his encouraging words when I needed them. Infinitely patient, he was always happy to have me bump into his office with yet another “petite question”. Thanks so much Jean-Philippe, I wouldn’t be here if it weren’t for you!

I would also like to thanks Prof. Murat Kunt, for helping me build a solid background in image processing and for welcoming me to the Signal Processing Institute (ITS), where I was able to work towards my thesis in impeccable conditions.

Next, I would like to thank Prof. Aude Billard, the president of my thesis exam, and the members of the jury, Dr. Xavier Bresson, Prof. Nikos Paragios and Prof. Pierre Vandergheynst, for all the attention they dedicated to my thesis and for their favorable and constructive remarks with respect to my work. Moreover, thank you Pierre for your feedback following my MMM presentations, which helped clarify the general terms of my thesis. Also, thank you Xavier for passing me the fever of variational methods and showing me the ropes in this fascinating field. I also owe a great part of my success to Nikos. Thank you for your warm welcome during my stay in Paris, for sharing your scientific insight with me and for constantly encouraging me to publish in the best conferences and journals in the field. I was also especially fond of the coffee breaks together with your research group in Paris, which combined the deepest philosophical discussions with mouth-watering desserts.

During my conference stays, I particularly appreciated many inspiring discussions with Prof. Alfred Bruckstein, Prof. Daniel Cremers, Prof. Nir Sochen and Prof. Demetri Terzopoulos and therefore I would like to thank them all here.

I would also like to thank Prof. David Barber and Dr. Bertrand Mesot for fueling my interest into graphical models. In particular, thank you Bertrand for all the time you invested into proof-reading my thesis, for your much-needed corrections and for your encouragement during the writing of my thesis.

I am particularly grateful to Leila Cammoun, my office mate, for her kindness and support, and for being a friend that I could count on in so many occasions. A small example of Leila’s selflessness is the fact that one day before my private thesis exam, despite a severe migraine, she came all the way to the EPFL to help me rehearse the thesis presentation. Thank you Leila, you are an angel!

Next, I would like to thank the whole Computer Vision group, for the feedback regarding my work, for having the patience to listen to my thesis presentation rehearsal, and especially for helping me improve it.

I would also like to thank the SP Labs group (former ITS) for making my stay at the EPFL a rich and unforgettable experience. I particularly enjoyed all the ski-days spent together with the group — thanks to the organizers, drivers and all participants, in particular Dan Jurca and Mathieu Lemay, who took care of the audio part (!). Moreover, thanks to all the discofit girls — Meri Bach, Nawal Houhou, Anna Llagostera and Francesca De Simone — for helping me stay fit and motivated for sports, and for sharing some bad muscle aches :). Thank you Karin Schnass for bringing energy to the group and for all your initiatives, and especially for kidnapping me from my birthday party.

I would like to thank Marianne Marion and Rosie de Pietro for helping me out with administration issues, which sometimes get pretty complicated at the EPFL. I am also very grateful to Gilles Auric for his prompt support on technical matters and especially for volunteering to take me sailing at lunch time on his wonderful boat!

I would particularly like to thank Radu Jurca and Carla Neumann for their much-needed help with practical issues when I moved to Switzerland, and especially for the moral support they offered me through my doctoral years. A special thank you is addressed to Oana Jurca, for without her I wouldn't have come to Switzerland and the EPFL. Thank you Oana for pushing me to apply to the doctoral program at the EPFL, but more importantly for being such a good friend and support over all these years. I would equally like to thank Răzvan Ludvig for showing me how to stay cool and for his invaluable sense of humor.

I would also like to thank all the members of the Romanian gang, inside and outside EPFL, for all the memorable moments spent together and for helping me miss home less. I am particularly grateful to the all the addicts of the EPFL lunch group, lead by Diana Ciressan and Bogdan Lazăr, for their toleration of EPFL food, but mostly for all the engaging conversations that helped take my mind off work.

A great thanks goes out to all my "dancing" friends, in particular Carolina and Carine Carvalho, and Charlotte Maisonneuve, for going dancing with me and for all the wonderful times we had together.

Moreover, I would like to thank Jos and Danny Levy for welcoming me at their place during so many weekends, for being great cooks and especially for cheering me up while I was struggling with my thesis.

A very special thank you is dedicated to Ron Levy, for always believing in me, for his support and patience during hard times, like the printing of my thesis at late night hours at the EPFL, and most importantly for making me happy and for all the love that he has been giving me.

My last thought is addressed to my parents. Surely I wouldn't be here without their constant support, encouragement and love, that have accompanied me at every step of life. Thanks infinitely!

Abstract

Visual behavior recognition is currently a highly active research area. This is due both to the scientific challenge posed by the complexity of the task, and to the growing interest in its applications, such as automated visual surveillance, human-computer interaction, medical diagnosis or video indexing/retrieval. A large number of different approaches have been developed, whose complexity and underlying models depend on the goals of the particular application which is targeted. The general trend followed by these approaches is the separation of the behavior recognition task into two sequential processes. The first one is a feature extraction process, where features which are considered relevant for the recognition task are extracted from the input image sequence. The second one is the actual recognition process, where the extracted features are classified in terms of the pre-defined behavior classes. One problematic issue of such a two-pass procedure is that the recognition process is highly dependent on the feature extraction process, and does not have the possibility to influence it. Consequently, a failure of the feature extraction process may impair correct recognition.

The focus of our thesis is on the recognition of single object behavior from monocular image sequences. We propose a general framework where feature extraction and behavior recognition are performed jointly, thereby allowing the two tasks to mutually improve their results through collaboration and sharing of existing knowledge. The intended collaboration is achieved by introducing a probabilistic temporal model based on a Hidden Markov Model (HMM). In our formulation, behavior is decomposed into a sequence of simple actions and each action is associated with a different probability of observing a particular set of object attributes within the image at a given time. Moreover, our model includes a probabilistic formulation of attribute (feature) extraction in terms of image segmentation. Contrary to existing approaches, segmentation is achieved by taking into account the relative probabilities of each action, which are provided by the underlying HMM.

In this context, we solve the joint problem of attribute extraction and behavior recognition by developing a variation of the Viterbi decoding algorithm, adapted to our model. Within the algorithm derivation, we translate the probabilistic attribute extraction formulation into a variational segmentation model. The proposed model is defined as a combination of typical image- and contour-dependent energy terms with a term which encapsulates prior information, offered by the collaborating recognition process. This prior information is introduced by means of a competition between multiple prior terms, corresponding to the different action classes which may have generated the current image. As a result of our algorithm, the recognized behavior is represented as a succession of action classes corresponding to the images in the given sequence.

Furthermore, we develop an extension of our general framework, that allows us to deal with a common situation encountered in applications. Namely, we treat the case where behavior is specified in terms of a discrete set of behavior types, made up of different successions of actions, which belong to a shared set of action classes. Therefore, the recognition of behavior requires the estimation of the most probable behavior type and of the corresponding most probable succession of action classes which explains the observed image sequence. To this end, we modify our initial model and develop a corresponding Viterbi decoding algorithm.

Both our initial framework and its extension are defined in general terms, involving several free parameters which can be chosen so as to obtain suitable implementations for the targeted applications. In this thesis, we demonstrate the viability of the proposed framework by developing particular implementations for two applications. Both applications belong to the field of gesture recognition and concern finger-counting and finger-spelling. For the finger-counting application, we use our original framework, whereas for the finger-spelling application, we use its proposed extension. For both applications, we instantiate the free parameters of the respective frameworks with particular models and quantities. Then, we explain the training of the obtained models from specific training data. Finally, we present the results obtained by testing our trained models on new image sequences. The test results show the robustness of our models in difficult cases, including noisy images, occlusions of the gesturing hand and cluttered background. For the finger-spelling application, a comparison with the traditional sequential approach to image segmentation and behavior recognition illustrates the superiority of our collaborative model.

Keywords: behavior recognition, variational image segmentation, probabilistic temporal model, Hidden Markov Model, Viterbi decoding, gesture recognition.

Résumé

La reconnaissance visuelle du comportement est un domaine de recherche très actif. Cela est dû à la fois au défi scientifique posé par la complexité de la tâche, et à l'intérêt croissant de ses applications, telles que la surveillance visuelle automatisée, l'interaction homme-machine, le diagnostic médical ou l'indexation / recherche automatique de vidéos. Un grand nombre d'approches différentes ont été développées, dont la complexité et les modèles sous-jacents dépendent des objectifs de l'application particulière envisagée. La tendance générale suivie par ces approches est la séparation de la tâche de reconnaissance du comportement en deux processus séquentiels. Le premier est un processus d'extraction de caractéristiques, où les caractéristiques qui sont considérées comme pertinentes pour la tâche de reconnaissance sont extraites de la séquence des images d'entrée. Le second est le processus de reconnaissance, où les caractéristiques extraites sont classifiées en fonction des classes de comportement prédéfinies. Un aspect problématique de ces procédures en deux passes est que le processus de reconnaissance est fortement dépendent du processus d'extraction de caractéristiques, et ne dispose pas de la possibilité de l'influencer. Par conséquent, une performance médiocre du processus d'extraction de caractéristiques peut empêcher la reconnaissance correcte de la séquence.

L'objectif de notre thèse porte sur la reconnaissance du comportement d'un objet unique dans une séquence d'images monoculaire. Nous proposons un cadre général où l'extraction de caractéristiques et la reconnaissance du comportement sont réalisées conjointement. Cela permet aux deux tâches d'améliorer mutuellement leurs résultats grâce à une collaboration et un partage des connaissances existantes. Cette collaboration est atteinte par l'introduction d'un modèle probabiliste temporel, basé sur un modèle de Markov caché (MMC). Dans notre formulation, le comportement est décomposé en une série d'actions simples et chaque action est associée à une probabilité différente d'observer un ensemble particulier de caractéristiques de l'objet dans l'image à un moment donné. En outre, notre modèle comprend une formulation probabiliste de l'extraction des attributs (caractéristiques), en termes de segmentation d'images. Contrairement aux approches existantes, la segmentation est obtenue en tenant compte du rapport des probabilités de chaque action, qui sont fournis par le MMC.

Dans ce contexte, nous résolvons le problème conjoint d'extraction des attributs et de reconnaissance du comportement par l'élaboration d'une variation de l'algorithme de décodage de Viterbi, adaptée à notre modèle. Dans la dérivation de l'algorithme, nous traduisons la formulation probabiliste d'extraction des attributs dans un modèle variationnel de segmentation. Le modèle proposé est défini comme une combinaison de termes d'énergie typiques basés sur l'image et sur le contour avec un terme qui résume l'information préalable, fournie par le processus de reconnaissance. Cette information préalable est introduite par le biais d'une compétition entre plusieurs termes, correspondant aux différentes catégories d'action qui auraient pu généré l'image courante. À la suite de notre algorithme, le comportement reconnu est représenté comme une succession de classes d'actions, correspondant aux images dans la séquence donnée.

En outre, nous développons une extension de notre cadre général, qui nous permet de traiter une situation souvent rencontrée dans les applications. Plus précisement, nous traitons les cas où le comportement est spécifié en termes d'un ensemble discret de types de comportement, et un tel type est défini par une succession différente d'actions, qui appartiennent à un ensemble de classes d'action. Par conséquent, la reconnaissance du comportement exige l'estimation du type de comportement le plus probable et de la succession correspondante de classes d'action la plus probable, qui explique la séquence image observée. À cette fin, nous modifions notre modèle initial et nous développons un algorithme de décodage de Viterbi adéquat.

Notre cadre initial, ainsi que son extension, sont définis en termes généraux, impliquant plusieurs paramètres libres qui peuvent être choisis de manière à obtenir l'implémentation appropriée pour l'application visée. Dans cette thèse, nous démontrons la viabilité des approches proposées en développant des implémentations particulières pour deux applications. Ces deux applications appartiennent au domaine de la reconnaissance de gestes et concernent le comptage sur les doigts et la dactylologie, respectivement. Pour la tâche de comptage sur les doigts, nous utilisons notre cadre initial, alors que pour l'application en dactylologie, nous utilisons son extension. Pour les deux applications, nous choisissons des modèles et des quantités particuliers pour les paramètres libres des deux cadres. Ensuite, nous expliquons l'entraînement des modèles obtenus à partir de données spécifiques d'entraînement. Enfin, nous présentons les résultats obtenus en testant nos modèles entraînés sur des nouvelles séquences d'images. Ces résultats montrent la robustesse de nos modèles dans des cas difficiles, notamment des images bruités, des occlusions de la main et des arrière-plans encombrés. Pour l'application en dactylologie, nous effectuons une comparaison avec l'approche traditionnelle séquentielle pour la segmentation d'images et la reconnaissance du comportement, qui montre la supériorité de notre modèle collaboratif.

Mots-clés: reconnaissance du comportement, segmentation variationnelle d'images, modèle probabiliste temporel, modèle de Markov caché, décodage de Viterbi, reconnaissance de gestes.

Contents

Acknowledgments **iii**

Abstract **v**

Résumé **vii**

Contents **ix**

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Image Segmentation and Behavior Recognition Towards Computer Vision . 1
1.2 Goal and Motivation of This Thesis 4
1.3 Main Contributions and Thesis Organization 8

2 Background and State of the Art **11**

2.1 Image Segmentation Using Variational Methods 11
  2.1.1 Edge-Based Active Contours 14
  2.1.2 The Level Set Method 17
  2.1.3 Region-Based Active Contours 24
  2.1.4 Introducing Prior Knowledge into Active Contours 33
  2.1.5 Object Tracking with Variational Segmentation Methods 43
2.2 Behavior Recognition with Computer Vision Approaches 45

3 A framework for Collaborative Segmentation and Behavior Recognition from Image Sequences **51**

3.1 Introduction 52
3.2 General Description of the Framework 54

3.3 A Model for the Joint Segmentation and Behavior Recognition of Image Sequences . . . 56
  3.3.1 Motivation . . . 56
  3.3.2 The Model . . . 56
3.4 Joint Segmentation and Behavior Recognition in Image Sequences . . . 60
3.5 Variational Segmentation Formulation for Contour Estimation . . . 64
3.6 Formal Derivation of our Competition-Between-Priors Scheme . . . 67
3.7 On the Benefits of Our Strategy for Competition Among Multiple Priors . . 70
3.8 Learning the Parameters of Our Model . . . 73
3.9 Extension of Our Framework for the Recognition of a Predefined Behavior Set 75
3.10 Summary . . . 79
3.11 Conclusion . . . 80

4 Applications and Solutions Implementing our Framework 83
4.1 Introduction . . . 84
4.2 A Finger-counting Application . . . 87
  4.2.1 Task Description . . . 87
  4.2.2 Implementation Based on Our Framework . . . 87
  4.2.3 Training the Model . . . 89
  4.2.4 Experimental Results . . . 92
4.3 A Finger-spelling Application . . . 93
  4.3.1 Application Description . . . 93
  4.3.2 Solution Based on the Proposed Framework . . . 94
  4.3.3 Management of Co-articulation Effects . . . 98
  4.3.4 Database and Training of the Model . . . 103
  4.3.5 Experimental Results . . . 103
4.4 Conclusion . . . 115

5 Conclusion 117
5.1 Achievements . . . 117
5.2 Discussion and Future Work . . . 118

A Appendix 121
A.1 The Minimization of Functionals Using the Calculus of Variations and Gradient Descent . . . 121
A.2 Image Segmentation Using the Gaussian Prior Model, for the Finger-counting Application . . . 123
  A.2.1 Evolution Equations . . . 123
  A.2.2 Numerical Approach . . . 124
A.3 Image Segmentation Using the PCA-based Prior Model, for the Finger-spelling Application . . . 125
  A.3.1 Evolution Equations . . . 125

A.3.2 Numerical Approach 126

Bibliography **129**

Index **141**

Curriculum Vitae **143**

List of Figures

1.1 Image representation with intensity values given by height. 2
1.2 Image representation with intensity values given by brightness. 3
1.3 Images that can be segmented based on intensity and texture. 5
1.4 Images which pose challenges to automatic segmentation. 6

2.1 Example of marker particle curve representation. 19
2.2 Simulation of water droplet reunion in an emulsion. 19
2.3 Topological change during contour evolution modeled via the level set method. 22
2.4 Two solutions for the propagation of a concave curve with normal speed $\beta = 1$. 23
2.5 Image segmentation with the "active contours without edges" model proposed by Chan-Vese [30]. 28
2.6 Image segmentation using the piecewise-smooth model proposed by Vese-Chan [149]. 30
2.7 Image segmentation using the geodesic active regions model proposed by Paragios [110]. 31
2.8 Image segmentation using the global minimization of the active contour model proposed by Bresson et al. [24]. 33
2.9 Segmentation of corpus callosum images using the model of Leventon et al. [89]. 35
2.10 Segmentation of airplane image with missing edges using the model of Tsai et al. [143]. 35
2.11 Segmentation of an ultrasound epicardium image using the model of Chen et al. [32]. 36
2.12 Segmentation of low-quality football player image using the model of Rousson and Paragios [126]. 37
2.13 Segmentation of an image affected by clutter using the model of Cremers et al. [43]. 38
2.14 Tracking of a rotating rabbit through clutter and occlusion, using the model of Cremers et al. [42, 44]. 39
2.15 Comparison of different dissimilarity measures proposed by Cremers and Soatto [41]. 39

2.16 Segmentation of a partially occluded walking person with the model of Cremers et al. [45, 48]. 40
2.17 Segmentation of the left brain ventricle in a MRI image with the method proposed by Bresson et al. [23]. 42
2.18 Segmentation of multiple familiar objects using multiple competing shape priors, using the model of Cremers et al. [45, 49]. 43
2.19 Segmentation of a noisy sequence of a walking person with the method proposed by Cremers [40]. 45

3.1 The task of object behavior recognition from image sequences. 54
3.2 The typical approach to behavior recognition, composed of two sequential steps: attribute extraction and classification. 54
3.3 Our framework for behavior recognition, based on the cooperation of the two processes: attribute extraction and classification. 55
3.4 The Dynamic Bayesian Network supporting our joint segmentation / behavior recognition framework. 57
3.5 Our DBN for the recognition of a predefined behavior set. 76

4.1 Samples from the four gesture classes that we use in our finger-counting application. 87
4.2 Segmentation with our collaborative model of two finger-counting image sequences in the presence of occlusion, background complexity and noise. Comparison with conventional segmentation. 90
4.3 Finger-counting image sequence: behavior recognition results plotted per frame. 93
4.4 Manual alphabet of the French-speaking part of Switzerland. 94
4.5 Two image sequence acquisition scenarios for our finger-spelling application. 95
4.6 Sample images and corresponding letter classes from training sequences used in our finger-spelling application. 104
4.7 Correct segmentation and behavior recognition using our framework, for a test sequence of the word “Albania”. Erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. 105
4.8 Correct segmentation and behavior recognition using our framework, for a test sequence of the word “Belarus”. Erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. 106
4.9 Correct segmentation and behavior recognition using our framework, for a test sequence of the word “Denmark”. Erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. 107
4.10 Correct segmentation and behavior recognition using our framework, for a test sequence of the word “Ecuador”. Erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. 108

4.11 Correct segmentation and behavior recognition using our framework, for a test sequence of the word “Estonia”. Erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. 109
4.12 Initialization process for the first frame of the “Belarus” sequence. 111
4.13 Partial classification results for the “Belarus” sequence. 112
4.14 Examples of erroneous classification using our method. 113

List of Tables

4.1 Vocabulary of our finger-spelling application. 94
4.2 Confusion matrix and recognition results for our finger-spelling application. 114

Introduction

1

1.1 Image Segmentation and Behavior Recognition Towards Computer Vision

Humans use their vision to recognize, understand or appreciate the surrounding world. Computer vision aims at making computers see. It draws its roots from an ancient dream of humankind, dating back to antiquity, that is the development of intelligent machines. More concrete steps towards this dream have been initiated with the emergence of the artificial intelligence field [128], in the middle of the 20th century. Its ultimate ambition is to equip computers with capabilities to solve problems and achieve goals in the world as well as (or even better than) humans do. One of the key ingredients for achieving this would be enabling computers to sense the world. Computer vision is dealing with the sense of sight. It is concerned with building tools that would allow computers to perceive and understand the world using digital images.

Vision may seem a very natural thing for humans, but for a computer this task is challenging due to several reasons. One of the main difficulties arises from the fact that our surrounding world is three-dimensional (3D), whereas the images that are available to the computer are generally two-dimensional (2D). The projection to two dimensions causes great information loss. Another problem can be observed if we look at the image represented in Fig. 1.1. The intensity values corresponding to each image location are represented on the vertical axis. It is unlikely that we would understand what this image represents; at least, not before looking at its equivalent — and more common — representation in Fig. 1.2. This shows that without using our a priori knowledge about the world (which looks more like Fig. 1.2 does to us), we wouldn't be able to tell that this image represents a child. What

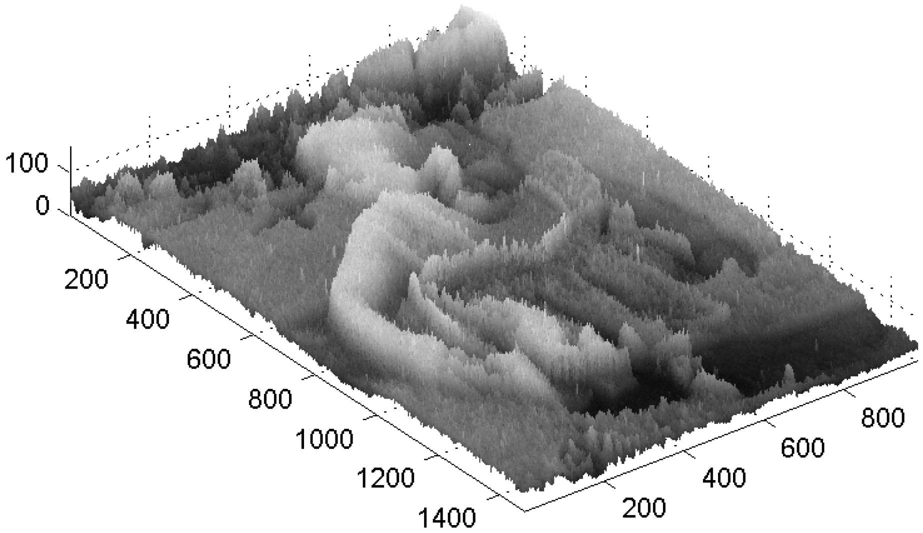


Figure 1.1 — An image representation where the intensity values corresponding to each image location are illustrated in terms of height (on the vertical axis).

we see in Fig. 1.1 is just the data that the computer receives about this image: a 2D array of numbers. From such local information it is difficult to find a global interpretation in the absence of supplementary knowledge. We can thus see that general purpose machine vision is hard to achieve, since the necessary extra-knowledge differs depending on the particular type of scenes/images that the machine needs to process.

Even though the dream of the ultimate intelligent (and seeing) machine is still quite far from realization, the field of computer vision has been steadily expanding, both at the theoretical level and through well-targeted applications in various fields. The automatization of the vision task has been motivated by cases where the volume of data is too large for a human to deal with in reasonable time (e.g. in medicine, for preliminary diagnosis, or in surveillance tasks, for event detection), where human attention and precision would diminish during long, repetitive tasks (e.g. in industry, for visual inspection), where the presence of a human operator would be impractical or dangerous (e.g. exploration by autonomous vehicles) or simply when communication with the computer through visual methods is desired (human-computer interaction).

To solve the vision problem, the first attempts tried to understand and reproduce biological vision. In particular, David Marr developed a general theory explaining vision [98], based on the idea that an exhaustive reconstruction of the visual environment is needed. This theory greatly influenced the computer vision field in its early years (1980s), but subsequently the field evolved towards more pragmatic approaches, focusing on particular vision subtasks, which were easier to model and solve using a computer. Nowadays, com-

Figure 1.2 — Common image representation where the intensity values corresponding to each image location are illustrated by brightness.

puter vision research and applications still benefit from biology-inspired techniques, while incorporating various methods and models from disciplines such as mathematics, physics, pattern recognition, artificial intelligence and computer science.

Typically, a complicated computer vision problem, which involves scene understanding (such as event detection in a surveillance application) is divided into several tasks, which are traditionally classified into two categories: low-level tasks and high-level tasks. Suppose that an image of the world has been acquired using a sensor (e.g. a camera), then digitized (for non-digital cameras), resulting in a 2D array of numbers which represent the brightness levels at each scene location, as projected into the image. Of course, for color cameras/images, three such arrays would result, corresponding to the three color channels: red, green and blue (RGB). Towards the resolution of the vision problem, first the low-level tasks would be performed. These are generally attributed to the image processing field and do not use much knowledge about image content. A typical sequence of such tasks would begin with image preprocessing, including steps such as noise removal, contrast enhancing, edge extraction or other operations which should emphasize key features for the understanding of the image (to be defined depending on the given application). The next task would be image segmentation, aiming at extracting relevant objects from the image, that should help its interpretation. This translates to outlining the image regions corresponding to these objects, either as groups of pixels or by their delimiting contours. Following segmentation, the delineated objects would typically be described in terms of a few key characteristics, such as their area, position, shape or average brightness. Next, the high-level vision tasks would be performed. These typically use techniques from the pattern recognition and artificial intelligence fields, in order to obtain an interpretation of the image. This can include the identification of the objects present in the image, of their current state or of the actions that they are currently involved in. The defining characteristic of high-level vision tasks is the use of a priori knowledge about the scene which is being visualized, such as the expected number of objects, their relevant characteristics and/or their expected behavior. In this way, image interpretation can be translated into a pattern recognition problem, where one searches for the most likely explanation of the object configuration detected by low-level vision, in terms of the set of a priori available hypotheses.

The focus area of this thesis is vision from image sequences. More concretely, given an image sequence of an object exhibiting a certain behavior, our aim is to delineate the object and to identify the respective behavior. In other words, we would like to perform segmentation of the image sequence and to recognize the behavior of the delineated object. As explained before, segmentation is typically regarded as a low-level vision task, whereas behavior recognition is a high-level vision task. In this thesis, we introduce a general framework which allows us to combine these two low- and, respectively, high-level tasks in a cooperative effort throughout image sequences.

1.2 Goal and Motivation of This Thesis

The main goal of this thesis is to find a joint solution to the problems of segmentation and behavior recognition from image sequences. Segmentation is a low-level task of computer vision, which aims at extracting meaningful objects from images [*]. We intend to fuse this task with a higher level vision task, which is the recognition of the behavior exhibited by the object in each image, based on prior knowledge about typical object behavior.

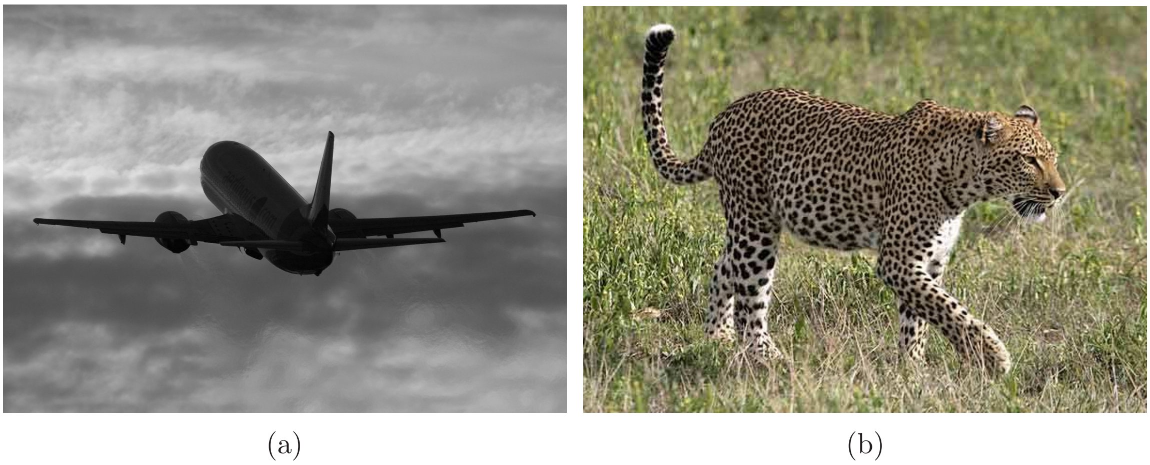
Defining our terms, by "behavior" we mean the temporal evolution of the object, as observed in the image sequence. Object behavior recognition refers to the interpretation of behavior as a succession of basic actions, each belonging to one of several possible action classes. Thus, the recognition of object behavior from an image sequence requires the determination of the appropriate action class throughout the sequence, for each object evolution instance. Behavior recognition can be used, for instance, to understand a sequence of object motions (e.g., car turn directions at an intersection), motions and deformations (e.g., hand gestures, body motions), or a sequence of intensity changes in a brain activation map for diagnostic purposes.

Classically, behavior recognition is formulated as a classification problem in terms of a series of relevant attributes (e.g., color histogram, object position, orientation, shape, size, etc.), which have been extracted from the image sequence in a preceding phase. Thus, the phase of attribute extraction, which may or may not involve image segmentation, is conventionally performed separately from behavior recognition. This thesis pursues a joint,

---

[*] Note that in this work we only consider the case of a single object of interest which evolves within an image sequence. Nevertheless, extensions of our work to deal with several objects are conceivable.

collaborative solution to the problems of image segmentation and object behavior recognition.

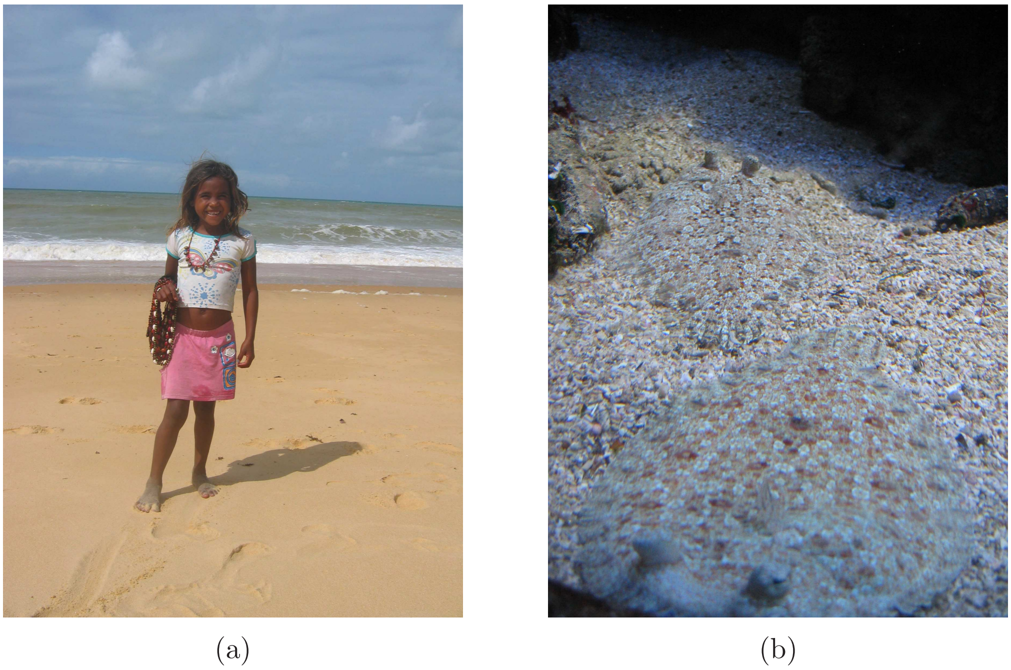


(a) (b)

Figure 1.3 — Examples of images that can be segmented based on intensity (a) and texture (b).

To motivate our thesis, we first explain why one would want to use high-level a priori knowledge for image segmentation. Depending on the type of images targeted for segmentation, the discrimination between the object of interest and the background can rely on different image-based characteristics, such as edge information, intensity levels, color or texture. For instance, to segment the image in Fig. 1.3(a), we would use the fact that the aeroplane is darker than the background. The leopard in Fig. 1.3(b) would be segmented based on its specific texture characteristics which distinguish it from the background. But sometimes such low-level information is not sufficient to correctly delineate the desired object. This is usually because the object and background do not (entirely) respect the fundamental assumption of segmentation, that is, that object and background can be correctly discriminated based on the chosen characteristics (intensity, color, texture, etc.). For instance, even though it is an obvious task for a human observer, for a machine it is quite challenging to segment the little girl in Fig. 1.4(a). This is because her appearance is not uniform, but made up of patches of different colors, and therefore cannot be succinctly described in terms of one characteristic. Moreover, the color of the girl's blouse is similar to that of the waves (background), which makes it difficult to accurately separate her from the background. Another image which would be difficult to segment automatically is presented in Fig. 1.4(b). Looking at the image, we can quite easily distinguish the two flatfish that are camouflaged in the sand. To achieve this, our brain uses cues from the image (the eyes of the fish, slight texture differences) and integrates them with prior knowledge about what fish look like. Therefore, it is a sensible decision to perform segmentation by combining the low-level information given by the image (intensity, color, texture) with higher level a priori knowledge regarding the expected characteristics of the target object(s) in the image (e.g. shape or defining landmarks).

Behavior recognition is a higher level task than segmentation. Nevertheless, it is en-



(a) (b)

Figure 1.4 — Examples of images which pose challenges to automatic segmentation (a) The appearance of the little girl is not uniform and partly confounds with the background. (b) The two flatfish hiding in the sand are hard to identify based solely on colour/texture.

tirely dependent on the results of low-level processes (such as segmentation) which extract relevant attributes from the images. Therefore, if these processes fail to capture the right attributes, behavior recognition will also be prejudiced. However, in order to perform behavior recognition, one must possess some form of extra knowledge (acquired a priori), regarding the possible types of exhibited behavior. Motivated by these considerations, in this thesis we will use a priori knowledge about behavior types to guide image sequence segmentation. Thus, we introduce a feedback loop between the two processes — image segmentation and behavior recognition — which helps them collaborate towards commonly improving their results. Our approach is derived by a natural analogy with the mechanism of human vision. In order to realize visual perception and understanding, the human brain blends prior knowledge, acquired through learning, with the immediate stimuli of the surrounding world.

Our work is also motivated by previous promising results towards the same direction, encountered in the literature. Many of these results belong to the field of variational image segmentation, which is also at the basis of our thesis. Variational segmentation offers us a principled, mathematically sound way of integrating different image-based segmentation criteria (edges, intensity, color, texture) and also higher level prior knowledge about the target object(s) (e.g. shape information, expected trajectory, etc). Significant contributions to the field were made by the introduction of models such as the active contours (snakes) [81], the Mumford-Shah model [103], geodesic active contours [28, 82, 95] and, more recently, versatile segmentation approaches such as [112, 149]. The use of prior shape information for the segmentation of familiar objects has been thoroughly elaborated in works such as [126], [44], [23], [48] and [49]. In particular, the use of shape priors has been successfully advocated for single images, in single-class [126], [23] and multiple-class [49] scenarios. To guide tracking over image sequences, [40] demonstrates the use of single-class dynamic models of motion and deformation, based on auto-regressive modeling. The novelty of our approach is that we fuse variational segmentation *over image sequences* with the problem of behavior recognition, and solve it in a *multi-class* scenario, i.e., where the behavior class of the tracked object changes over time.

Another incentive for our work is the fact that it provides a new outlook on the issue of behavior recognition. The vast majority of approaches to behavior recognition regard it as a problem of classification, formulated in terms of time-series of attributes a priori extracted from image sequences. To give a few examples, Bobick and Davis [17] extract Hu moments of motion history images and classify new movements based on the shortest Mahalanobis distance to learned models of each action. Schüldt et al. [132] detect local features based on space-time image gradients and recognize actions using Support Vector Machines (SVMs). Gorelick et al. [70] obtain space-time shapes as concatenations of segmented 2D human silhouettes and extract various shape properties, which they use for action representation and classification. Many behavior recognition systems use tracking in some form, be it of entire objects, object parts or of some relevant features [*]. Thus, we observe that most existing behavior recognition methods consist of an initial phase, where relevant attributes are extracted from the image sequences, based on various criteria, and of a second phase, where the behavior reflected in the image sequence is classified using the extracted attributes and knowledge gathered from training data. This means that in the first phase (attribute extraction) some information is automatically discarded, without considering higher level knowledge which could be obtained from the existing training data. Also, the retained attributes could be affected by low image quality (noise, occlusions) or poor separation of the target object(s) from the background. In this thesis, we propose a novel approach to the behavior recognition problem, which relies on the collaboration between the low-level attribute extraction process (performed through image segmentation) and the higher level behavior recognition process. This allows the existing knowledge relevant to each of the processes to aid in the resolution of the other. Therefore, the common information is better exploited to the benefit of both processes and of their final result.

Last, but not least, our work is stimulated by the wealth of applications for both image segmentation and behavior recognition. These cover a multitude of domains, such as medicine (e.g., for automatic diagnostics), sports (e.g. for the improvement of athletic performance), car industry (e.g. sleep detection, pedestrian detection), multimedia (e.g. video annotation and compression), surveillance (e.g. unusual behavior detection, shopping behavior analysis) or human-computer interfaces. In particular, to demonstrate the feasibility of our proposed general framework for segmentation and behavior recognition, we have fo-

---

[*] Comprehensive surveys of the work regarding tracking and recognition of human behavior can be found in [64, 101].

cused on two gesture recognition applications: finger-counting and finger-spelling. These will be detailed in Chapter 4.

1.3 Main Contributions and Thesis Organization

The main contributions of our thesis can be summarized as follows:

1. **A general collaborative framework for segmentation and behavior recognition from image sequences.** Our framework relies on the collaboration between the high-level behavior recognition process and a low-level attribute extraction process, based on variational image segmentation. This collaboration is based on a mutual exchange of information, which is beneficial for both tasks. Our framework is derived by formulating the joint segmentation / recognition problem in terms of a Dynamic Bayesian Network. Recognition and segmentation are naturally derived in terms of probabilistic inference via a variation of the Viterbi decoding algorithm. This enables the interleaving of the two processes along the image sequence, and the intended collaboration. In particular, classification offers dynamic probabilistic priors to guide segmentation, while segmentation supplies its results to classification, ensuring that they are consistent with prior knowledge. Our general framework can be employed in solving a wide range of applications, by adapting its components and parameters according to the specific need.

2. **An extension of our framework to deal with the recognition of a predefined behavior set.** Our original framework from **1.** yields the recognized behavior in terms of its composing succession of action classes, that correspond to the frames of an image sequence. We extend this framework to the recognition of behaviors belonging to a predefined set of behavior types. These behavior types are characterized by different successions of action classes.

3. **A variational segmentation model based on dynamical statistical priors.** We develop a new variational segmentation model for image sequences. It is derived from an initial probabilistic formulation, stemming from our collaborative approach for segmentation and behavior recognition. It consists of the classical image- and contour-based terms and of a term which encapsulates prior knowledge, offered by the collaborative recognition process. The latter term guides the segmentation of each image towards the most likely location of the target object, based on learning from the training data and on reasoning about future behavior, on account of the evidence gathered up to the present moment. Based on considerations flowing from our probabilistic formulation, we search for the object belonging to the most probable class at the moment and thus introduce the dynamical priors offered by each action class in a competition approach. We show that our strategy for introducing this competition outperforms a well-known method in the literature in terms of the fairness with respect to all the priors involved.

4. **A concrete model for collaborative segmentation and behavior recognition, based on Gaussian priors.** With the purpose of solving a proposed finger-counting recognition application, we instantiate our general framework from **1.** with specific segmentation and probabilistic models. In particular, we employ level-set-based active contours [108] as the extracted image attributes and we use the piecewise-constant Chan-Vese model [30] to fill in the image- and contour-based terms for segmentation. Attribute probabilities for each action class are given by a local Gaussian model of the level set function, motivated by [126]. Our resulting implementation offers robust segmentation and recognition results, for cases of difficult images, affected by clutter, noise and occlusions of the target object.

5. **A concrete model for collaborative segmentation and behavior recognition, using PCA-derived prior contours.** We are targeting a finger-spelling recognition application. Finger-spelling is the part of sign language which consists of manual representations of alphabet letters. In collaboration with the Swiss Federation for the Hearing Impaired, we acquired a database of finger-spelt words, that we consequently use to train and test our segmentation/behavior recognition framework. To solve our application, we derive a concrete implementation of our extended framework from **2.**. It is based on the same attribute (the level set representation of the hand contour) and image- and contour-based energy terms as the model at **4.**. To allow more flexibility in the modeling of prior information, we use prior contours based on Principal Components Analysis (PCA) for each action class, in an approach motivated by [23]. The attribute probability for each action class is modeled in terms of a distance function with respect to the prior contour of the respective action class. To improve computational efficiency and convergence towards the correct action class, we propose a pruning mechanism which reduces the number of priors competing for the guiding of segmentation. The obtained segmentation and recognition results show the ability of our approach to deal successfully with complicated backgrounds, as well as its superiority with respect to the traditional sequential approach, which separates segmentation and recognition.

This dissertation is organized as follows:

Chapter 2 presents the theoretical background of this thesis. In particular, the first section of the chapter introduces the state-of-the-art models used for variational image segmentation. We start by a general presentation of variational segmentation approaches and then describe edge-based active contours. Next, we present the level set method for contour representation and continue with region-based active contours — some of which will be employed in the concrete implementations of our framework, in Chapter 4. Segmentation models which make use of prior knowledge regarding the segmented objects are presented at the end of the first section. They prepare the inclusion of dynamical prior information into our variational segmentation model described in Chapter 3. The second part of Chapter 2 presents a few general considerations regarding behavior recognition and introduces some state-of-the-art models in the literature.

Chapter 3 introduces our main contribution: the general framework for collaborative segmentation and behavior recognition from image sequences, as well as its extension dealing with a predefined behavior set. At first, we show the motivation behind our framework and give its general description. Then, we present the Dynamic Bayesian Network which enables our joint treatment of the segmentation and recognition problems. We develop a probabilistic inference scheme based on Viterbi decoding, which results in the interleaved, collaborative resolution of segmentation and recognition along the image sequence. In particular, from our probabilistic model, we derive a variational segmentation method based on the competition between multiple priors, offered by the recognition process. Then, we extend our framework to deal with the case of a set of predefined behavior types, each decomposable into a succession of action classes. Finally, we summarize our approach by a concrete step-by-step description.

Chapter 4 illustrates the potential of our general framework through two applications, regarding finger-counting and finger-spelling recognition, respectively. For each application, we explain the task and describe the available data. Then, use we our general framework and its extension for a predefined behavior set (described in Chapter 3) to derive concrete implementations aimed at the respective applications. Finally, we present the segmentation and recognition results that we have obtained by our collaborative approach and compare them with results obtained by the traditional approach, which separates the two phases of segmentation and recognition.

Chapter 5 concludes our thesis by summarizing our achievements and discussing directions for future work.

Background and State of the Art

2

In this chapter, we set the theoretical background of our thesis. In the first part of the chapter, we introduce the variational approach to image segmentation and present the state of the art in the field. First, we consider the earliest models, based on the detection of image edges. Then, we present a valuable tool, extensively used in recent variational segmentation approaches and also in our thesis: the level set method for contour representation. We continue by presenting variational segmentation formulations which employ region-homogeneity criteria, including the Mumford-Shah functional and its cartoon simplification, that we utilize in our work. Next, we review a few representative models which introduce prior shape knowledge into variational segmentation, and finally we report on the use of variational segmentation for object tracking. We thus set the basis for one of our own contributions, consisting of a variational segmentation model which incorporates dynamical attribute priors and is used for tracking the target object over image sequences.

In the second part of the chapter, we treat computer vision approaches to behavior recognition. In particular, we outline a few significant directions adopted in the field, together with associated publications. At the same time, we explain where our own work stands with respect to these approaches and emphasize our original contributions.

2.1 Image Segmentation Using Variational Methods

Image segmentation is one of the most basic yet most challenging problems of computer vision. Segmentation requires finding in an image semantically salient regions (or their bounding contours), which correspond to objects or areas of the real world captured in the image. More formally, the problem can be described as the process of partitioning the image plane $\Omega \subset \mathbb{R}^2$ into a set of non-overlapping regions $\{\Omega_i\}_{i=1..N}$, corresponding to the meaningful image structures:

$$\bigcup_{i=1}^{N} \Omega_i = \Omega, \qquad \Omega_i \cap \Omega_j = \emptyset \quad \text{if} \quad i \neq j.$$

This kind of segmentation is know as “strong segmentation” [102]. Alternatively, one can aim solely for the separation of one or several objects of interest from the image background.

Segmentation is a long standing problem in computer vision and numerous different approaches have been developed. A review of the whole field is beyond the scope of our thesis. The interested reader is referred to the book of Sonka et al. [138], which offers a broad overview of the segmentation techniques, including the thresholding approach, methods based on edge detection, region growing and the watershed approach. In the following, we will focus on variational methods for image segmentation, which are a fundamental component of our thesis.

Variational methods underlie the mathematical formulation of numerous computer vision problems. In a variational approach, the solution to these problems is obtained as an optimizer of an energy functional, which encapsulates a number of relevant constraints for the given problem. To give an example, let us consider the problem of noise removal from an image. Intuitively, noise is made up of small artifacts in an image, which impair the observation of the interesting objects. In general, the scale of these artifacts is smaller than that of the interesting objects and this is one important criterion for their elimination. The other one is that technically they are manifested as fluctuations of pixel intensity with respect to the surrounding locations. Denoting the noisy image by $I_0 : \Omega \to \mathbb{R}^+$, in a variational approach a de-noised image $I : \Omega \to \mathbb{R}^+$ can be obtained by minimizing the following energy functional:

$$E(I) = \int_{\Omega} (I(x,y) - I_0(x,y))^2 dx\, dy \; + \lambda^2 \int_{\Omega} |\nabla I(x,y)|^2 dx\, dy. \tag{2.1}$$

This functional is a simple example from a general class of variational formulations that can be used to solve a variety of image processing problems, as shown by Terzopoulos in [140]. Such functionals are made up of two terms: a fidelity term (in our case $\int_\Omega (I(x,y) - I_0(x,y))^2 dx\, dy$), which measures how faithful the approximation I is to the original data I_0, and a regularization term (in our case $\int_\Omega |\nabla I(x,y)|^2 dx\, dy$), which measures how smooth the approximation I is. For our de-noising problem, the smoothness is measured by integrating the square magnitude of the intensity gradient over the image domain. The balance between the two terms is dictated by the weighing parameter λ^2, also called a scale parameter, since it controls the amount of smoothing of I and thus the minimum scale of details that will be kept in the smoothed image I.

To obtain our de-noised image $I^* = \min_I E(I)$, with $E(I)$ defined by (2.1), we use the calculus of variations and gradient descent, yielding the following evolution equation for I:

$$\frac{\partial I(x,y,t)}{\partial t} = I(x,y,t) - I_0(x,y) + \lambda^2 \Delta I(x,y,t).^* \tag{2.2}$$

---

[*] For better readability, from now on we will omit the image coordinates (x, y) and artificial time t from

Here t is an artificial time-marching parameter and Δf stands for the Laplacian of the function f: $\Delta f(x, y) = f_{xx}(x, y) + f_{yy}(x, y)$ [*]. Starting with the initial condition $I(x, y, 0) = I_0(x, y)$, the evolution equation (2.2) is iterated until a steady state is reached, and thus the solution I^* is obtained. The minimization of functionals by the calculus of variations and gradient descent is briefly presented in the Appendix of this thesis, Section A.1. The obtained partial differential equation (PDE) (2.2) is known as the linear diffusion equation, which, as shown by Koenderink [83], is equivalent to convolving the image with a Gaussian function at a certain scale. As a result, all image structures at the respective scale will be smoothed, including the edges of the target objects. In order to eliminate noise without perturbing the relevant image structures, one can turn to nonlinear diffusion (for more details see for instance [102, 154]).

There are several advantages to using variational formulations for image segmentation, compared to the use of other existing algorithms [102, 130]. First of all, contrary to most other methods, which treat images in a discrete setting (i.e., as 2D arrays of numbers), variational methods model images as functions defined on a continuous domain (as you can notice in our example above). In this way, the formalism becomes grid-independent and isotropic, and is amenable to study and development via continuous mathematics, which are better developed than discrete mathematics. Concrete implementations of variational methods, applicable to digital images, can be obtained by discretizing the resulting PDEs using efficient tools from numerical analysis.

Another advantage of variational methods is the fact that they condensate all the criteria regarding the desired segmentation into a single functional. This functional has a real value, which allows the evaluation of the quality of a specific segmentation (with respect to the defined criteria) and also the comparison of different segmentations. In other words, there is a single quantity which needs to be optimized, which can incorporate a multitude of different criteria in a mathematically sound way. This will allow us to perform segmentation in a principled way by integrating dynamical attribute priors offered by the behavior recognition process. This elegance and ease of use make the variational approach more appealing than many heuristic segmentation methods, which are usually sequences of different steps, dependent on one another and involving the empirical choice of many parameters. Moreover, as shown in [102], most existing segmentation methods can be formulated within a variational approach, i.e., as a functional whose optimization yields the desired solution. This enables a clear expression of the involved parameters and also possible reductions of their number.

Image segmentation using variational approaches is usually performed by deforming one or more contours (also known as active contours) within the image domain, in order to minimize a given energy functional. This functional can incorporate several criteria/objectives which guide the active contour towards the desired segmentation. The first generation of

---

our equations, whenever this does not impair understanding. For instance, $I(x, y, t)$ will be denoted simply as I.

[*] We use the subscript notation to denote partial derivatives with respect to spatial coordinates, whenever it doesn't hinder clarity.

variational methods for segmentation relied on attracting the contour towards regions of high intensity gradient in the image (image edges). Since they depend on gradients, these methods are very sensitive to noise and poor image contrast. The robustness of these methods was increased by the introduction of more global information into the segmentation functional, concerning region homogeneity, which was modeled statistically. Moreover, segmentation in difficult cases, including occlusions of the target object(s) and cluttered background, was improved by the integration of more specific a priori information about the target object(s), regarding its shape or predicted position.

In the following, we will present a few representative models from all three categories of variational segmentation models: edge-based, region-based and including a priori information. This will support the development of our own variational segmentation model in Chapter 3 and will help us differentiate our approach with respect to the existing ones. Additionally, we will present the level set method for contour representation, that we employ in the resolution of two applications in Chapter 4. A classical method for functional minimization, which supports the implementation of many variational segmentation approaches developed in the literature, is succinctly described in the Appendix of this thesis.

2.1.1 Edge-Based Active Contours

Edges are locations of intensity discontinuity within an image. Their importance for vision lies in the fact that they normally correspond to discontinuities in scene geometry or reflectance, therefore indicating the separation between different objects or areas. Edge detection is an essential component of biological vision, as first demonstrated by R. von der Heydt et al. [152].

By analogy with biological vision, many low-level computer vision algorithms aim at the extraction of edges from images. Technically, edges in an image correspond to locations of high intensity gradient, or alternatively to zero-crossing locations of the Laplacian of image intensity. The fact that edge-detectors are local operators, based on image gradients, makes them very sensitive to noise. Moreover, the definition of edges involves a scale factor: intensity discontinuities can only be detected at a certain spatial scale. Both issues are addressed by the theory of multiscale filtering and multiscale edge detection, thoroughly studied in the literature (see for example [26, 79, 83, 90, 99, 118, 154, 156]). Basically, the image targeted for edge detection is first smoothed at different scales, using for instance the linear diffusion exemplified at the beginning of this section, or a suitable nonlinear filter [154]. This creates a fine-to-coarse family of images, known as the scale space of the original image. Afterwards, edges are detected as maximum locations of the gradient (or Laplacian zero-crossings), measured on the image smoothed at a certain scale, which gives the scale of edge detection.

Once the edges of an image have been detected, the remaining task for segmentation is to link them into a coherent representation (usually by line-drawing) of the objects/regions present in the image. Early segmentation methods attempted to perform edge linking by using heuristics which relied on edge strength, length and alignment, involving the empirical setting of different parameters [125, 155].

An elegant solution, which simultaneously addresses the two problems of edge detection and edge linking, was the original variational formulation of active contours (also known as "snakes"), introduced by Kass, Witkin and Terzopoulos [81]. This formulation is based on deforming an initial contour towards locations of interest within the image (such as lines or edges). The deformation is obtained by the minimization of an energy functional, which is designed so that its local minima matches these locations of interest.

The snake is defined as a parameterized planar curve $C(p) = (x(p), y(p)) \in \Omega$, where $x(p)$ and $y(p)$ are the x, y coordinates along the contour and $p \in [0, 1]$. It deforms in order to minimize the energy functional

$$E(C) = \int_0^1 E_{int}(C(p)) + E_{image}(C(p)) + E_{con}(C(p))dp, \tag{2.3}$$

where E_{int} represents the internal energy of the curve, E_{image} designates image forces and E_{con} imposes external constraints, dictated by the user or by another high-level process. Usually, the snake is represented as a spline, in order to ensure continuity properties. Based on this original formulation of Kass et al. [81], the most popular choice of snake model, which also lies at the basis of future approaches, is given by:

$$E(C) = -\int_0^1 |\nabla I(C(p))|^2 + \alpha \int_0^1 |C_p(p)|^2 + \beta \int_0^1 |C_{pp}(p)|^2. \tag{2.4}$$

The first term is an edge-based term, which drives the snake towards locations of high image gradient (edges). The next two terms introduce a smoothness constraint: the first of them makes the snake resist stretching (its integral gives the contour length), while the second makes it resist bending. These properties can be controlled by adjusting the weighing factors α and β. The minimization of energy (2.4) via the calculus of variations and gradient descent yields the curve evolution equation

$$\frac{\partial C}{\partial t}(p) = \nabla |\nabla I(C(p))|^2 + \alpha C_{pp}(p) - \beta C_{pppp}(p). \tag{2.5}$$

The fourth order derivative in this equation is difficult to approximate in a discrete setting and is the source for numerical instability. Therefore, in practice, β is often set to zero.

One important weakness of the snake model is its local character: it is only driven by image features which lie in its near vicinity, and therefore can fail to capture the desired object if initialized too far from it. An improvement with respect to this issue is brought by the introduction of the so-called "balloon" model [33]. This model applies an additional force to the contour, which makes it behave like a balloon which is inflated/deflated. In this way, the contour becomes more dynamic, being able to escape spurious local minima and thus locate image features further away from its initial position. The disadvantage of this approach is that the snake looses generality: one has to know a priori the direction of the applied force, i.e. whether the snake needs to shrink or to expand in order to reach the desired object.

Another limitation of the snake model is the fact that the functional (2.3) depends on the curve parametrization p. This parametrization is only related to the velocity at which the curve is traveled, and not to the intrinsic geometry of the curve. A change in the parametrization p can change the resulting energy. Thus, one can obtain different minimization results for one and the same initial contour, which is an undesirable effect. The solution to this problem was offered by the model of geodesic active contours [27, 28, 82]. To develop this model, Caselles et al. [28] start from the original snake model (2.4), excluding the second-order smoothness term ($\beta = 0$):

$$E(C) = -\int_0^1 |\nabla I(C(p))|^2 + \alpha \int_0^1 |C_p(p)|^2. \tag{2.6}$$

The edge-detection term, depending on the intensity gradient $|\nabla I(C(p))|$, is generalized by the introduction of an edge-detecting function $g : [0, +\infty[\to \mathbb{R}^+$, strictly decreasing and vanishing at infinity: $g(x) \to 0$ when $x \to \infty$. Thus, $-|\nabla I|^2$ is replaced with $g(|\nabla I|)^2$, resulting in:

$$E(C) = \int_0^1 g(|\nabla I(C(p))|)^2 + \alpha \int_0^1 |C_p(p)|^2. \tag{2.7}$$

The role of the function g is to stop contour evolution on object edges. A popular variant of such function is:

$$g(|\nabla I|) = \frac{1}{1 + \gamma |\nabla I^s|^p}, \tag{2.8}$$

where I^s is a Gaussian-smoothed version of the original image I, γ is a positive constant and $p \in [1, 2]$. A detailed analysis regarding the selection of $g(|\nabla I|)$ can be found in [55].

Following the introduction of the generalized edge-detecting function, the authors of [28] show that the minimization of (2.7) is equivalent to the minimization of the functional:

$$E(C) = \int_0^{L(C)} g(|\nabla I(C(s))|)\, ds = \int_0^1 g(|\nabla I(C(p))|)\, |C_p(p)|\, dp. \tag{2.9}$$

Here $ds = |C_p(p)|dp$ is the Euclidian arc length and $L(C) = \oint |C_p(p)|dp = \oint ds$ is the Euclidian length of the curve C, hence the last equality of equation (2.9). We notice that energy (2.9) is obtained from the Euclidian curve length, by weighing the Euclidian arc length ds by $g(|\nabla I(C(s))|)$, which indicates edge locations in the image. Caselles et al. [28] show that the minimization of this new length is equivalent to finding a geodesic curve (a curve of minimum distance) in a Riemannian space, whose metric tensor is determined by the image I. This property led to the appellation of the model (2.9) as geodesic active contours.

The minimization of (2.9) via the calculus of variations and gradient descent leads to the curve evolution equation [28]

$$\frac{\partial C}{\partial t} = g(|\nabla I(C)|)\, \kappa \mathcal{N} - (\nabla g(|\nabla I(C)|) \cdot \mathcal{N}) \mathcal{N}. \tag{2.10}$$

Here κ is the Euclidian curvature * and $\mathcal{N}$ is the unit inward normal to the curve. The first term of the right-hand side represents a curve shortening flow, weighed by the edge-detecting function g. The curve shortening flow (also known as the mean curvature motion) is the curve evolution $\frac{\partial C}{\partial t} = \kappa \mathcal{N} = C_{ss}$, which achieves the minimization of the curve length functional: $L(C) = \oint |C_s(s)| ds = \oint ds$. This flow has the double effect of smoothing the curve and of reducing its length. The weighing by $g(|\nabla I(C)|)$ slows down curve evolution on object boundaries. The second term of (2.25) has the role of attracting and fixing the curve to the middle of object boundaries (due to ∇g, which points from both sides of the boundary towards its central section). Therefore, the function g does not need to be equal to zero (the case of an ideal edge) to stop the curve evolution on object boundaries.

As we have seen, the geodesic active contours offer an elegant solution to the parametrization problem of the original snake model. Another important drawback of the latter formulation is the fact that it cannot deal naturally with topological changes of the contour (such as splitting or merging). This means that if an initial contour is surrounding several objects, the segmentation will not be able to capture these objects separately, since the contour topology is fixed. Partial relief to this problem was brought by methods which explicitly deal with contour merging/splitting, at the cost of additional complications (e.g. [86, 88, 100]). A definite answer to the problem was provided by the introduction of the level set method for curve representation and evolution, which will be presented in the following section.

2.1.2 The Level Set Method

The level set method is a technique for tracking moving interfaces, i.e. boundaries between two regions. Such interfaces exist in many different settings, including physical phenomena such as waves breaking, flames burning or different liquids blending, but can also be used to model various problems, such as optimal path planning or image segmentation. The main merits of the level set method result from the implicit representation of the interface geometry. This allows the automatic handling of topological transformations, such as region splitting or merging, as well the development of efficient and accurate numerical methods for practical implementations. Its versatility, together with its intrinsic qualities, have made the level set method a very popular theoretical and numerical tool in many fields, including physics, chemistry, fluid mechanics, materials sciences, combustion, seismology, computer graphics, image processing and computer vision. The level set method was originally proposed by Osher and Sethian in [108]. A light introduction to the topic is offered in [135], while detailed descriptions of the theoretical and numerical aspects can be found in [95, 106, 107, 136]. In the following, we will provide a brief outline of the level set method, in the context of curve/surface evolution.

---

[*] Intuitively, the curvature measures the bending speed of a curve. Technically, there exist several equivalent definitions for the curvature [130]. In terms of the parametric representation $C(p) = (x(p), y(p))$, the curvature is defined as $\kappa(p) = \dfrac{x_p y_{pp} - y_p x_{pp}}{(x_p^2 + y_p^2)^{3/2}}$, which in the case of the arc length parametrization becomes $\kappa(s) = x_s y_{ss} - y_s x_{ss}$.

We start by explaining the motivation for introducing the level set approach, arising from the difficulties posed by the existing curve evolution methods. If we consider a closed planar curve which is deforming in time, such a process would generate a family of curves $C(p,t) : [0,1] \times [0,T) \to \mathbb{R}^2$, where t parameterizes the family and p parameterizes the curve. The general curve evolution equation is [130]:

$$\frac{\partial C}{\partial t}(p,t) = \alpha(p,t)\,\mathcal{T}(p,t) + \beta(p,t)\,\mathcal{N}(p,t), \tag{2.11}$$

where $C(p,t=0) = C_0(p)$ is the initial condition. Here $\mathcal{T}$ stands for the unit tangent to the curve and $\mathcal{N}$ for the unit inward/outward normal (its direction can be arbitrarily chosen). The equation states that the curve is deforming with α velocity in the tangential direction and with β velocity in the normal direction. The tangential velocity does not affect the geometry of the deformation (i.e., how the curve looks), but only its parametrization (i.e., the speed at which one travels along the curve, by changing the parameter p) [60]. Since we are only interested in changing the geometry of the curve, the general evolution equation can be simplified to

$$\frac{\partial C}{\partial t}(p,t) = \beta(p,t)\,\mathcal{N}(p,t), \tag{2.12}$$

by eliminating the term which contains the tangential velocity. This result is also valid for the general case of interfaces between two regions, such as surfaces in $\mathbb{R}^3$ or hyper-surfaces in $\mathbb{R}^n$.

Now let us suppose that we would like to deform a curve so that each of its points moves in the normal direction, with a velocity dictated by its curvature (motion by mean curvature), according to the equation:

$$\frac{\partial C}{\partial t}(p,t) = \kappa(p,t)\,\mathcal{N}(p,t). \tag{2.13}$$

Such motion produces a relaxation of the curve, which becomes smoother and decreases its length, gradually becoming circular, before collapsing into a single point [71].

If we are to use a parametric representation as the basis for our numerical implementation, we would approximate the curve using a set of marker particles placed around the curve, tied together by continuity constraints (e.g., via a B-spline curve representation). Figure 2.1 presents a symbolic example of such a curve representation, where the red dots correspond to marker particles and the green arrows stand for normal direction velocities given by the motion by mean curvature (2.13). Of course, in practice more particles would be used for a more accurate curve description. This figure reflects some inconveniences of such a curve representation with respect to the intended curve motion. The marker particles have a tendency to cross each other's path, which makes it difficult to maintain their original organization. Moreover, as the curve continues to shrink, the particles crowd together along the diminished curve length, causing growing errors in the numerical approximation of the derivatives. A cumbersome solution would be to advance the curve by small steps and periodically re-sample the curve, setting new marker positions and decreasing their number according to the need.

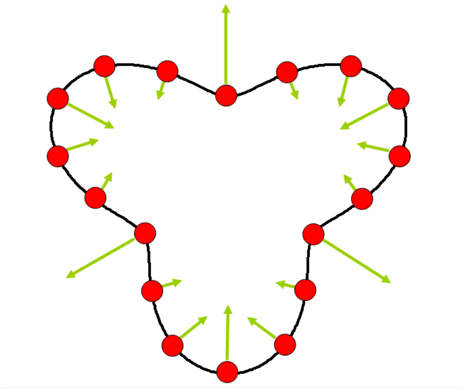
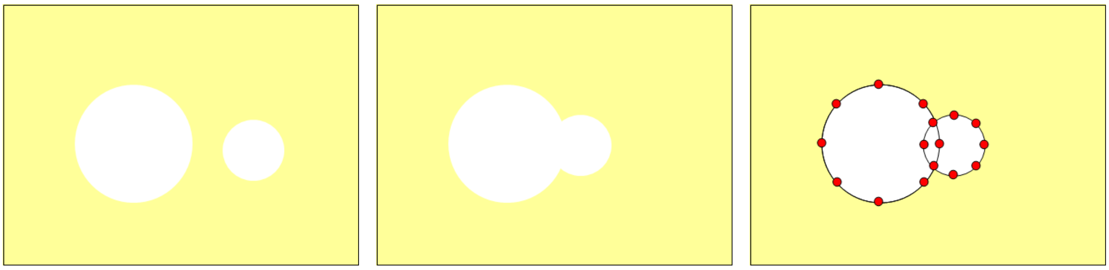


Figure 2.1 — Example of marker particle curve representation. The red dots stand for the marker particles. The green vectors represent the velocity in the normal direction obtained through motion by mean curvature.

Another problem becomes apparent when the deforming boundary needs to change its topology. For instance, imagine that we would like to simulate the reunion of two water droplets in an emulsion (Fig. 2.2 (a)). At some point, their boundaries touch and they merge into a single larger droplet (Fig. 2.2 (b)). If we use our parametric approach to track their boundaries, we end up in a situation such as the one illustrated in Fig. 2.2 (c). In order to correctly follow the intended evolution, we need to find a way to detect and eliminate the marker particles lying inside the merged region. Defining a general algorithm for removing such particles is a daunting task. As we will see in the following, all these problems are elegantly solved by using the level set method for curve representation.



(a) (b) (c)

Figure 2.2 — Simulation of water droplet reunion in an emulsion. (a) Droplets before reunion. (b) Droplets during reunion. (c) Marker particle model of reuniting droplets' boundaries.

The main idea of the level set approach is that instead of following the curve itself, one adds an extra dimension to the problem and follows the resulting surface. More formally, a closed interface $\Gamma \in \mathbb{R}^n$ (e.g. a curve in $\mathbb{R}^2$, a surface in $\mathbb{R}^3$ or a hyper-surface in $\mathbb{R}^n$ is represented as a level set of a higher dimensional function, called level set function. For example, in the case of a 2D curve, the embedding function would be a 3D surface $z = \phi(x, y)$ and the curve would be given by the set of points (x, y) which are at the same height (level) of ϕ: $\phi(x,y)=c$, with c a given constant [*]. Considering that the interface Γ is evolving in time, its level set representation ϕ is a scalar Lipschitz continuous function $\phi:\Omega\subset\mathbb{R}^n\times[0,T)\rightarrow\mathbb{R}$, which respects the conditions [107]:

$$\begin{cases} \phi(\mathbf{x},t)>0 & \text{for } \mathbf{x}\in\omega, \\ \phi(\mathbf{x},t)<0 & \text{for } \mathbf{x}\in\Omega\setminus\omega, \\ \phi(\mathbf{x},t)=0 & \text{for } \mathbf{x}\in\partial\omega, \end{cases} \tag{2.14}$$

where $\omega\subset\Omega$ denotes the region enclosed by the interface Γ and $\partial\omega$ denotes the boundary of ω. For example, to represent a planar curve, a common choice for the level set function is the signed distance function to the curve $d(x,y)$, with a positive sign in the interior of the curve and a negative sign in its exterior.

In this way, the evolution of a curve can be modeled through the evolution of its level set function. The position of the curve at any time can be retrieved as the (zero) level set of the embedding function ϕ: $C(t)=\{(x,y)|\phi(x,y,t)=0\}$. The geometric properties of the curve can be directly obtained from the level set function. The unit normal to a level set is given by

$$\mathcal{N}=-\frac{\nabla\phi}{|\nabla\phi|} \tag{2.15}$$

(the sign depends on the assumed direction of the normal). The curvature can be calculated as [†]

$$\kappa=-\mathrm{div}\mathcal{N}=-\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{2.16}$$

which is equivalent [130] to

$$\kappa=\frac{\phi_{xx}\phi_y^2-2\phi_{xy}\phi_x\phi_y+\phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}. \tag{2.17}$$

Moreover, the length of the curve and its enclosed area can be expressed by defining $\delta_\epsilon, H_\epsilon:\mathbb{R}\rightarrow\mathbb{R}$ as suitable smooth functions which, as $\epsilon\rightarrow 0$, approximate the Dirac distribution and the Heaviside function

$$H(z)=\begin{cases} 1 & \text{if } z\geq 0, \\ 0 & \text{if } z<0 \end{cases}, \tag{2.18}$$

respectively, while having $\delta_\epsilon=H'_\epsilon$. Then, the length of the curve is given by

$$L_\varepsilon(\phi)=\iint_\Omega|\nabla H_\varepsilon(\phi)|dxdy=\iint_\Omega\delta_\varepsilon(\phi)\,|\nabla\phi|\,dx\,dy, \tag{2.19}$$

and the area of its enclosed region ω by

$$A_\varepsilon(\phi)=\iint_\Omega H_\varepsilon(\phi)\,dx\,dy. \tag{2.20}$$

---

[*] For convenience, one usually chooses to embed the interface as the zero level set of the hyper-surface.

[†] We denote by div the divergence of a vector $\mathbf{v}=(v_1,v_2)$: $\mathrm{div}\,\mathbf{v}=\frac{dv_1}{dx}+\frac{dv_2}{dy}$.

Furthermore, from the condition that the level set function must be equal to a constant along the embedded curve $C(t)$, i.e.,

$$\phi(C(t), t) = c, \tag{2.21}$$

we can derive the evolution equation for the level set function ϕ which matches the evolution of $C(t)$ given by (2.12). Differentiating this condition with respect to the time t yields (by the chain rule):

$$\frac{\partial \phi}{\partial t}(C(t), t) + \nabla \phi(C(t), t) \cdot \frac{\partial C}{\partial t}(t) = 0. \tag{2.22}$$

Since $\dfrac{\partial C}{\partial t} = \beta \mathcal{N}$ and $\mathcal{N} = -\dfrac{\nabla \phi}{|\nabla \phi|}$, we obtain the following evolution equation for ϕ

$$\frac{\partial \phi}{\partial t} = \beta \, |\nabla \phi|, \tag{2.23}$$

with the initial condition $\phi(t = 0) = \phi_0$, where ϕ_0 is the level set function of the given initial curve C_0. Note that when the level set function is evolving according to equation (2.23), the level set corresponding to the embedded curve, as well as all the other level sets, deform according to the curve evolution equation (2.12).

Let us now look at the advantages offered by the level set method. First of all, this approach for curve representation is parameter-free, since it is written in a fixed coordinate system (x, y), as opposed to the parametric approach, which relies on a geometric, mobile coordinate representation. This means that we no longer need to adjust the parametrization to suit the curve configuration, as was the case for the parametric approach, where marker particles had to be managed and redistributed along the curve to accurately follow its evolution. Instead, with level sets we can track a curve simply by adjusting the height of the level set function in each point (x, y) of our domain. In particular, this also brings a great relief from the problem of topological changes, which are naturally handled in the level set framework, or, as Osher [107] would put it, with “no emotional involvement”. There is no need for developing intricate algorithms to track topological changes, since the topology of the level set function does not change. The merging or splitting of the underlying curves occur automatically with the evolution of the level set function and are discovered when the corresponding level set is computed. This property is illustrated in Fig. 2.3, which depicts the curve evolution for the segmentation of an image containing two triangles (right column). The initial contour is a circle (the red contour in the first image of the right column), which deforms to capture the shapes of the two triangles. This deformation naturally leads to its splitting (last row), which does not necessitate any additional effort and is produced by updating the level set function values according to the corresponding PDE.

Another problem which is well addressed in the context of level set methods is the development of sharp corners and discontinuities in the evolving interface. Such discontinuities can arise simply by deforming a curve via a PDE like (2.12), using a constant velocity in the normal direction $\beta = 1$. Taking the example of a concave initial curve, two solutions to such a propagation are illustrated in Fig. 2.4. These solutions are similar up to the appearance of a corner in the propagating interface. After that moment, one solution crosses

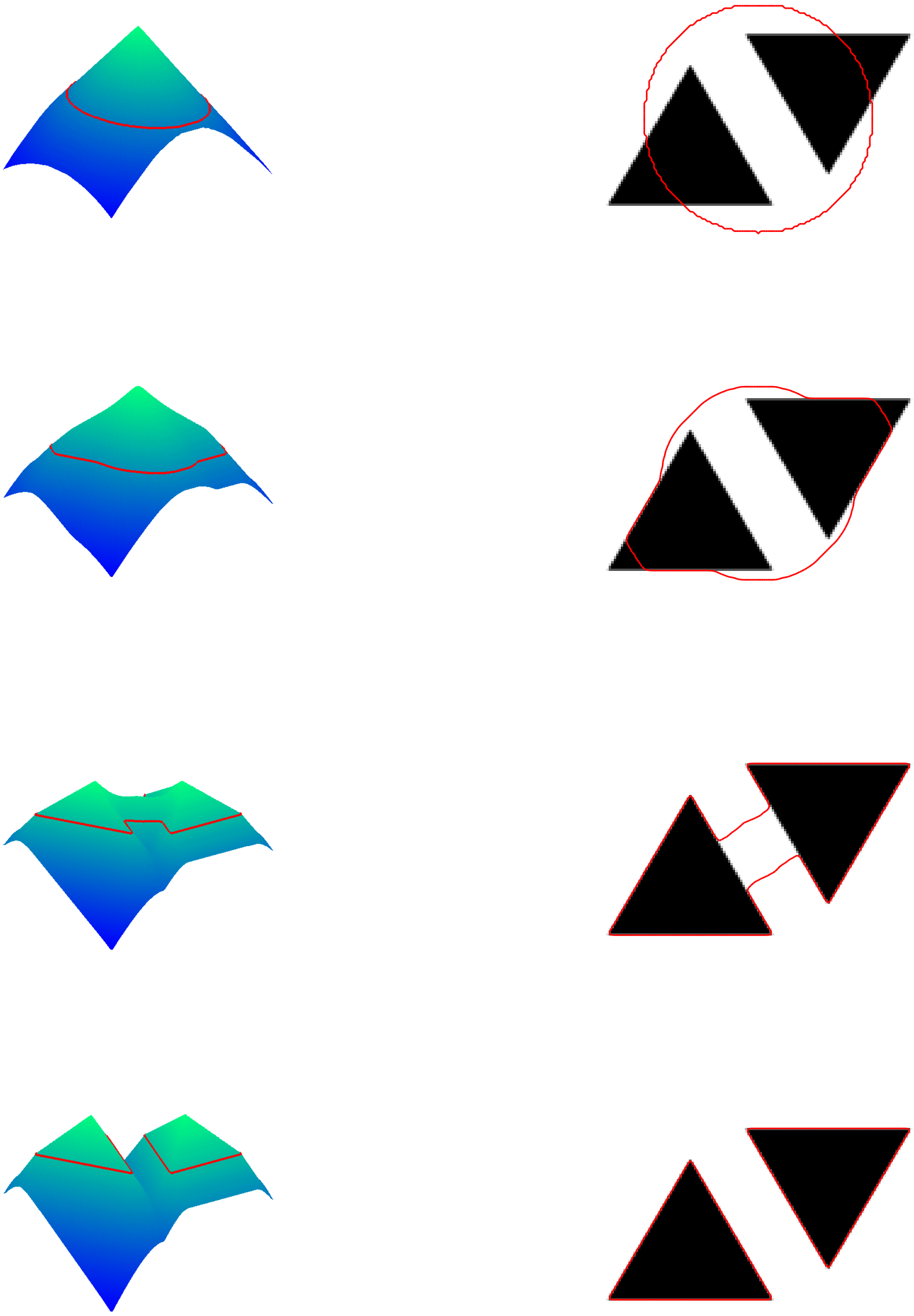


Figure 2.3 — Demonstration of a topological change during contour evolution modeled via the level set method. The evolution corresponds to the segmentation of the top image in the right column. The initial contour position is illustrated as the red contour in the top figure of the right column. Left column: evolution of the level set function, with the zero level set marked as the red contour. Right column: associated curve evolution (red contour), superimposed over the image targeted for segmentation.

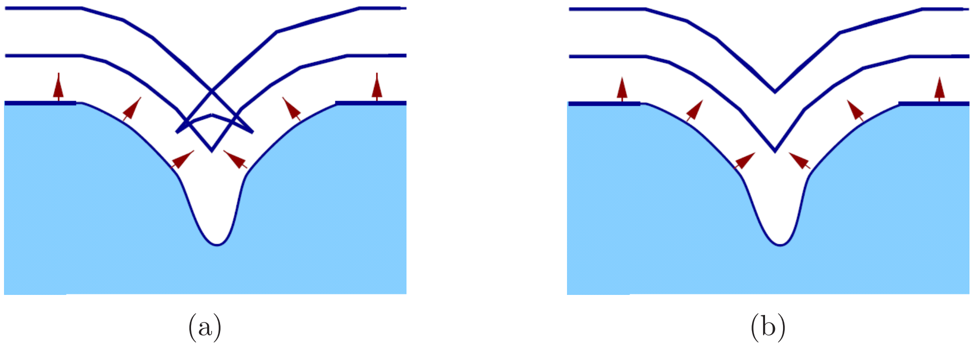

(a) (b)

Figure 2.4 — Illustration of two solutions for the propagation of a concave curve with normal speed $\beta = 1$. (a) Swallowtail solution. (b) Leading front solution. This figure is reproduced from [136].

over itself, whereas the other selects only the leading front. Intuitively, the leading front solution seems like the physically correct one.

The reason why several solutions to this problem are possible is the fact that once a corner develops in the solution, the normal to the interface is ambiguously defined and it is not obvious how to proceed with the evolution. Therefore, the only possibility is the calculation of a "weak solution", i.e. a solution which only weakly satisfies the definition of differentiability (see [135] for more details). Both illustrated solutions are weak solutions. The first one is obtained by continuing the motion of each individual point. The second one respects our intuition that all the points of the interface at a certain evolution step should be located at an equal distance from the interface position at the previous time step, since they advance with equal normal velocities. As shown in [133–135], the second solution, which is the physically correct one, can be obtained by imposing an "entropy condition", similar to the one used for hyperbolic conservation laws. The theoretical framework which allows us to obtain this entropy-satisfying weak solution is the mathematical theory of viscosity solutions, pioneered by Crandall et al. [36–38].

Returning to the level set evolution equation (2.23), we note that if the velocity β depends only on the position $\mathbf{x}$ and on first-order derivatives of ϕ, this is a particular case of the general Hamilton-Jacobi equation

$$u_t + H(Du, \mathbf{x}) = 0, \tag{2.24}$$

where Du designates the partial first-order derivatives of u in each variable and the Hamiltonian $H(Du, \mathbf{x}) = -\beta|\nabla u|$. The property of the level set evolution equation of being a Hamilton-Jacobi equation (in certain conditions) allows the use of the theory of viscosity solutions in order to obtain non-smooth solutions that allow corners (thus being non-differentiable), as in the previous example.

Regarding the numerical implementation of the resulting PDEs, the key idea is to borrow existing technologies for the numerical solution of hyperbolic conservation laws and apply them to the Hamilton-Jacobi setting. The principle behind the utilized numerical schemes is that "the numerical domain of dependence should contain the mathematical domain of dependence" [135]. This gives rise to the so-called "upwind schemes", where the computation of function values at the current grid point uses values upwind of the direction of information propagation. As a result, one can develop accurate and stable numerical algorithms, which yield physically reasonable, entropy-satisfying (and possibly discontinuous) solutions to the employed PDEs. For more details regarding numerical schemes, see [135].

In contrast to the level set approach, curve evolution methods based on the parametric representation, implemented by marker-particle techniques, are incapable to account for the proper entropy condition [135] and are affected by stability problems, being unable to cope with discontinuous solutions. In such methods, small errors due to the imprecision of marker positions accumulate and amplify uncontrollably through a feedback loop involving the computation of derivatives. Therefore, the use of impractically small time steps for curve evolution is required, together with mechanisms intended to keep the particles apart from each other. Apart from being complicated, such techniques modify the motion equations in non-definite ways, which is not desirable.

Summing up, all the presented theoretical and practical tools create a rigorous mathematical framework for the study and development of level-set-based curve evolution equations. This has encouraged the use of the level set method in many variational image segmentation approaches, including our own implementations for gesture recognition applications, presented in Chapter 4. For example, Caselles et al. [28] embedded the geodesic active contour evolution equation (2.25) into the evolution of a level set function ϕ, with velocity $\beta = g(|\nabla I|)\kappa - (\nabla g(|\nabla I|) \cdot \mathcal{N})$ for each level set, yielding (by (2.23)):

$$\frac{\partial \phi}{\partial t} = g(|\nabla I|)\,\kappa\,|\nabla \phi| - (\nabla g(|\nabla I|) \cdot \nabla \phi). \tag{2.25}$$

2.1.3 Region-Based Active Contours

In Section 2.1.1, we presented edge-based variational segmentation approaches, where the contour is attracted to the closest positions of locally maximum image gradient (edges). The main disadvantage of these approaches is their local character, which means that they cannot reach beyond the nearest energy minimum, and thus can become trapped into undesirable local minima caused by spurious edges/noise formations in the image. To encourage the convergence of these methods over larger distances and also make them more robust against insignificant local intensity variations, an image smoothing step was introduced prior to the edge detection via the function g (2.8). In turn, this creates a new inconvenient: the smoothing has the effect of smearing image edges and therefore exact information about their location is lost. Desirably, one would like a smoothing method which only eliminates noise artifacts and leaves object boundaries intact. However, prior to segmentation, we do not know where these boundaries are. This circular problem was addressed by the variational segmentation model of Mumford and Shah [103, 104].

The goal of the presented edge-based segmentation methods was the segmentation of a particular object of interest within an image. From a different perspective, the Mumford-Shah model aims at finding a strong segmentation of an image, by partitioning it into a set of disjoint homogenous regions. To this end, and in response to the problem explained above, it defines the segmentation problem as a coupled smoothing/edge detection problem: given an observed image I_0, find a piecewise smooth approximation I of I_0, featuring a set C of discontinuities, corresponding to the edges of I_0. These conditions are encapsulated by the Mumford-Shah energy functional:

$$E(I,C) = \iint_{\Omega} (I - I_0)^2 \, dx \, dy + \mu \iint_{\Omega \setminus C} |\nabla I|^2 \, dx \, dy + \nu |C|, \tag{2.26}$$

where $|C|$ stands for the one-dimensional Hausdorff measure of the length of C and $\mu > 0$, $\nu > 0$ are fixed parameters, weighting the energy terms. The first term is a fidelity term, imposing the similarity of the approximation I with the original image I_0. The second term constrains the image I to be smooth, with the exception of the set of discontinuities C. The last term is a regularization term which demands that the set C be of minimal length, and thus in particular as smooth as possible. Obviously, this simple functional cannot offer an accurate description of the complex structures encountered in most natural images. The approximated image I, together with the set of edges C, can only provide a simplified, cartoon-like representation of a scene. However, the better the target image matches the model assumption (i.e. is composed of piecewise-smooth object regions), the more satisfactory the segmentation result will be.

Since its apparition, the Mumford-Shah model has been the focus of attention of many theoretic studies and practical implementation efforts. Its importance is demonstrated (among others) in the book of Morel and Solimini [102], which shows that it can be considered the "general model of image segmentation, and all the other ones are variants, or algorithms tending to minimize these variants". Detailed analyses of the model can be found in the books of Morel and Solimini [102] and Aubert and Kornprobst [9]. Generally, theoretical studies of the model have been revolving around the question of existence of segmentations which minimize the Mumford-Shah functional, of their uniqueness and of the smoothness of the resulting boundaries. The theoretical minimization problem belongs to the class of free discontinuity problems [*] and is difficult due to the interaction of the two-dimensional terms in I and the one-dimensional length term. The existence of minimizers (generally not unique) was proved by De Giorgi and Ambrosio [4–6, 67, 68], while the regularity of the minimizing boundaries was demonstrated in [8, 18]. From a practical viewpoint, solutions for the direct computation of minimizers of the Mumford-Shah functional are not available. Nevertheless, a lot of research effort has been dedicated to approximating the functional with formulations that are feasible for numerical implementations [7, 15, 20, 29]. Level set implementations of the Mumford-Shah functional were proposed by Chan and Vese [30, 149] and by Tsai et al. [143].

The parameter μ of the Mumford-Shah functional (2.26) controls the amount of smoothing of the approximated image I. Increasing this parameter to the limit $\mu \to \infty$ results in the approximation I being piecewise-constant, i.e. constant in each region $\Omega_i \subset \Omega$ induced

---

[*] The term of "free discontinuity problem" was introduced by De Giorgi [66] and designates a variational problem involving a competition between volume energies, concentrated on N-dimensional sets, and surface energies, concentrated on $(N-1)$-dimensional sets, whose supports are not fixed a priori.

by the set of edges C:

$$I(x,y) = c_i \quad \text{for} \quad (x,y) \in \Omega_i. \tag{2.27}$$

In this case, the functional (2.26) is simplified to its "cartoon limit" [103]:

$$E(\{c_i\}, C) = \sum_i \iint_{\Omega_i} (I_0(x,y) - c_i)^2 \, dx\, dy + \nu |C|. \tag{2.28}$$

The minimization of this functional leads to a piecewise-constant approximation of the original image I_0 within the regions Ω_i determined by the boundaries C. For given C, the constants c_i are equal to the mean intensity values in each region Ω_i, as implied by imposing the minimum condition $\dfrac{dE}{dc_i} = 0$:

$$c_i = \frac{\iint_{\Omega_i} I_0(x,y)\, dx\, dy}{\iint_{\Omega_i} dx\, dy}. \tag{2.29}$$

Since these values are implied by the boundaries C, the functional (2.28) is only dependent on C, which simplifies its theoretical analysis and also its practical implementation.

One important merit of the piecewise-constant Mumford-Shah model (2.28) is that of opening the way for the introduction of probabilistic modeling into region-based variational segmentation approaches. In this context, we would like to mention the work of Zhu and Yuille on "region competition" [166]. Their approach combines region growing techniques with a variational approach, which is an extension of the piecewise-constant Mumford-Shah model to probabilistic region modeling. In the piecewise-constant Mumford-Shah model (2.28), the intensity value of each region Ω_i is approximated by the mean intensity value of the region. Zhu and Yuille propose a generalization of this representation: they describe the intensity of each region via a probabilistic model $P(I_0(x,y)|\alpha_i)$, where α_i denotes the parameters of the model. Then, the energy of each region Ω_i is given by the negative log-likelihood of observing an intensity value $I(x,y)$ at pixel (x,y), given the region model with parameters α_i:

$$E(\{\alpha_i\}, C) = \sum_i \left(- \iint_{\Omega_i} \log(I_0(x,y)|\alpha_i)\, dx\, dy + \frac{\nu}{2} \int_{\partial\Omega_i} ds \right). \tag{2.30}$$

Here $C = \cup_i \partial\Omega_i$ are the segmentation boundaries and the second term of the energy stands for the length of these boundaries. The piecewise-constant Mumford-Shah model can be regarded as a particularization of this formulation, by modeling the regions Ω_i with Gaussian probabilities of means c_i and constant variances.

The advantage of the Zhu-Yuille formulation is that it allows several useful extensions of the piecewise-constant Mumford-Shah functional. For instance, the authors themselves demonstrate the use of Gaussian probabilities, which allows the modeling of the regions Ω_i in terms of different means μ_i and variances σ_i:

$$P(I_0(x,y)|\alpha_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(- \frac{(I_0(x,y) - \mu_i)^2}{2\sigma_i^2} \right), \tag{2.31}$$

with $\alpha_i = \{\mu_i, \sigma_i\}$. This makes possible the segmentation of regions with similar means, but different variances. Furthermore, the Gaussian model permits straightforward extensions to vector-valued functions $I_0 : \Omega \to \mathbb{R}^n$. In particular, this allows the segmentation of images using color and texture information, as demonstrated in [166].

An implementation of the piecewise-constant Mumford-Shah model in the context of active contours represented by level sets is proposed by Chan-Vese in [30]. Their model is entirely region-based, hence the name of "active contours without edges". Starting from the assumption that the image is composed of two pixel classes (two phases), grouped into an inside region ω and an outside one $\Omega \setminus \omega$, Chan-Vese propose the energy functional

$$E(c_1, c_2, \omega) = \lambda_1 \iint_\omega (I_0(x,y) - c_1)^2 \, dx\, dy + \lambda_2 \iint_{\Omega\setminus\omega} (I_0(x,y) - c_2)^2 \, dx\, dy + \nu \text{Length}(\partial\omega), \tag{2.32}$$

with $\lambda_1, \lambda_2, \nu$ fixed positive constants, $c_1, c_2 \in \mathbb{R}$ and $\partial\omega$ representing the boundary of the region ω. The minimization of this functional yields the (locally optimal) L^2-norm approximation of the original image in terms of two region categories, characterized by two intensity values c_1 and c_2, under a length constraint over the regions' boundaries. As shown for the piecewise-constant Mumford-Shah model [103], for a fixed segmentation ω, the values c_1, c_2 minimizing (2.32) are given by the mean intensity values of the two regions ω and $\Omega \setminus \omega$.

For the implementation of their model, Chan-Vese use a level set formulation of (2.32):

$$\begin{aligned} E(c_1, c_2, \phi) = {} & \lambda_1 \iint_\Omega (I_0(x,y) - c_1)^2 H_\epsilon(\phi(x,y)) \, dx\, dy \\ & + \lambda_2 \iint_\Omega (I_0(x,y) - c_2)^2 H_\epsilon(-\phi(x,y)) \, dx\, dy \\ & + \nu \iint_\Omega |\nabla H_\epsilon(\phi(x,y))| \, dx\, dy. \end{aligned} \tag{2.33}$$

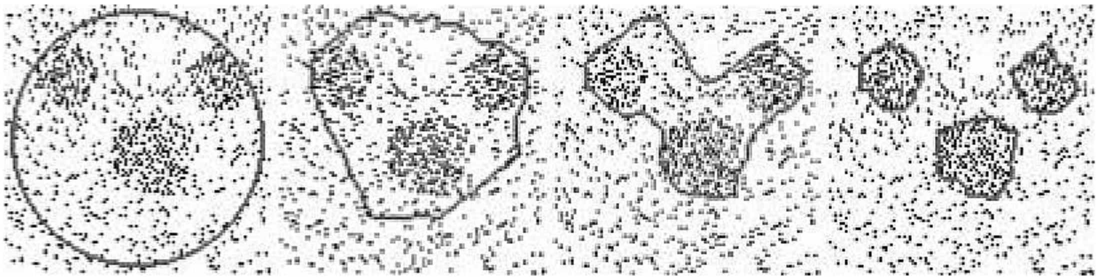
Here ϕ is the level set function embedding the boundary $\partial\omega$ and H_ϵ is a suitable smooth approximation of the Heaviside function (2.18), permitting differentiation. The Heaviside function is used as a characteristic function, allowing the discrimination of the two regions ω and $\Omega \setminus \omega$, which correspond to the positive and, respectively, negative regions of the level set function ϕ. The minimization of (2.33) is typically performed in two phases. First, ϕ is considered fixed and the energy is minimized with respect to c_1 and c_2, yielding:

$$c_1 = \frac{\iint_\Omega I_0(x,y) H_\epsilon(\phi(x,y)) \, dx\, dy}{\iint_\Omega H_\epsilon(\phi(x,y)) \, dx\, dy}, \qquad c_2 = \frac{\iint_\Omega I_0(x,y) H_\epsilon(-\phi(x,y)) \, dx\, dy}{\iint_\Omega H_\epsilon(-\phi(x,y)) \, dx\, dy}. \tag{2.34}$$

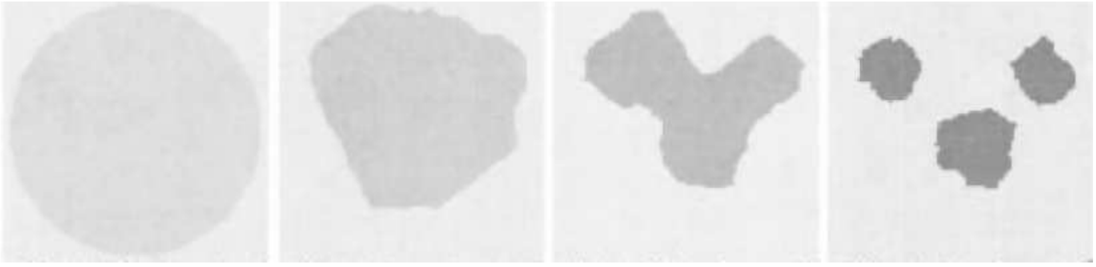
Then, c_1 and c_2 are considered fixed and the energy is minimized with respect to ϕ, yielding (by the Euler-Lagrange equation and gradient descent) the evolution equation

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left(-\lambda_1 (I_0(x,y) - c_1)^2 + \lambda_2 (I_0(x,y) - c_2)^2 + \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right). \tag{2.35}$$

Here the $\delta_\epsilon(\phi)$ designates the derivative of the Heaviside function $\delta_\epsilon(\phi) = H'_\epsilon(\phi)$ and is an approximation of the Dirac distribution, thus delimiting the zero-level set of ϕ. The



(a)



(b)

Figure 2.5 — Illustration of the capabilities of the "active contours without edges" model proposed by Chan-Vese [30]. (a) Segmentation of a simulated minefield, having a contour without edges. (b) Piecewise-constant image approximation resulting from the segmentation at (a). These images are reproduced from [30].

piecewise-constant approximation of the original image resulting from the segmentation can be computed as:

$$I(x,y) = c_1 H_\epsilon(\phi(x,y)) + c_2 H_\epsilon(-\phi(x,y)). \tag{2.36}$$

An example of segmentation which justifies the model name — "active contours without edges" — is illustrated in Fig. 2.5. The target image represents a simulated minefield where the regions cannot be distinguished based on edges, but rather in terms of their different dark-pixel density, which is reflected as a difference in mean intensity. The success of the Chan-Vese model is due to the use of a global segmentation criterion, which is evaluated by using information from the whole image (the mean intensity level), as opposed to the local gradient information, used by the edge-based methods.

A piecewise-smooth approximation of the original Mumford-Shah model (2.26) was proposed by Vese and Chan in [149]. Its main advantage with respect to the piecewise-constant model is the more faithful approximation of the input image (as a piecewise-smooth function). This translates to more accurate segmentations for classes of images that do not respect the piecewise-constant assumption, at the cost of increased complexity (and time costs) of the segmentation model. Considering a binary image partition (two pixel classes), the unknowns of the model are I — the piecewise-smooth approximation of the original image I_0 — and the set of boundaries $\partial\omega$ delimiting the two regions. The relation between these two unknowns can be expressed by introducing two functions I^+ and I^-, designating I on the inside and on the outside, respectively, of the boundaries $\partial\omega$. Formally, the following holds:

$$I(x,y) = \begin{cases} I^+(x,y) & \text{if } (x,y) \in \omega, \\ I^-(x,y) & \text{if } (x,y) \in \Omega \setminus \omega. \end{cases} \tag{2.37}$$

Using I^+ and I^-, the Mumford-Shah functional can be approximated as:

$$\begin{aligned} E(I^+, I^-, \omega) \; = & \iint_{\omega} (I_0(x,y) - I^+(x,y))^2 \, dx \, dy + \iint_{\Omega\setminus\omega} (I_0(x,y) - I^-(x,y))^2 \, dx \, dy \\ & + \mu \iint_{\omega} |\nabla I^+(x,y)|^2 \, dx \, dy + \mu \iint_{\Omega\setminus\omega} |\nabla I^-(x,y)|^2 \, dx \, dy + \nu \text{Length}(\partial\omega). \end{aligned} \tag{2.38}$$

In a level set formulation, (2.39) can be written as

$$\begin{aligned} E(I^+, I^-, \phi) \; = & \iint_{\Omega} (I_0(x,y) - I^+(x,y))^2 H_\epsilon(\phi(x,y)) \, dx \, dy \\ & + \iint_{\Omega} (I_0(x,y) - I^-(x,y))^2 H_\epsilon(-\phi(x,y)) \, dx \, dy \\ & + \mu \iint_{\Omega} |\nabla I^+(x,y)|^2 H_\epsilon(\phi(x,y)) \, dx \, dy \\ & + \mu \iint_{\Omega\setminus\omega} |\nabla I^-(x,y)|^2 H_\epsilon(-\phi(x,y)) \, dx \, dy \\ & + \nu \iint_{\Omega} |\nabla H_\epsilon(\phi(x,y))| \, dx \, dy. \end{aligned} \tag{2.39}$$

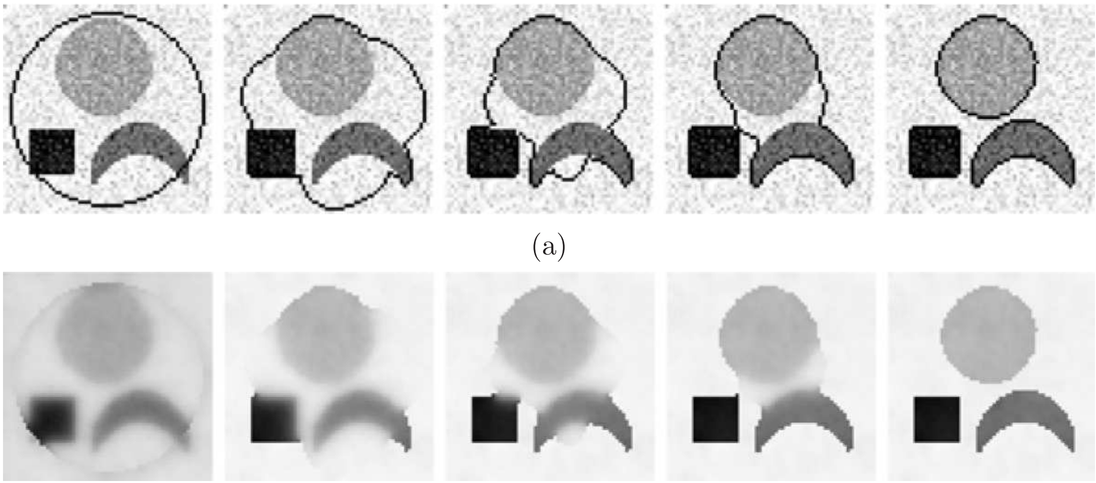
The minimization of this energy with respect to I^+, I^- and ϕ is achieved via the corresponding evolution equations, derived through the calculus of variations and gradient descent. The equations for the computation of the approximations I^+ and I^- lead to the smoothing and denoising of the original image I_0, in particular inside homogenous regions and avoiding the edges. The piecewise-smooth approximation of I_0 can be calculated as:

$$I(x,y) = I^+(x,y) H_\epsilon(\phi(x,y)) + I^-(x,y) H_\epsilon(-\phi(x,y)). \tag{2.40}$$

Figure 2.6 shows an example of segmentation which demonstrates the denoising capabilities of the piecewise-smooth Vese-Chan model. An advantage of using the level set implementation is that the contour splits automatically to capture all three objects present in the image.

The piecewise-constant and piecewise-smooth segmentation models described above are based on the assumption of a binary image partition (two pixel classes), which implies that the edges of the image can be described using a level set of a single level set function. The general case of an image partition which allows structures such as triple junctions, as well as the representation of multiple phases (classes) was treated by Zhao et al. [165], Vese-Chan [149] and Tsai et al. [143].

A variational segmentation framework which integrates edge- and region-based approaches is proposed by Paragios and Deriche in [110–113]. The proposed model, named "geodesic active regions", builds upon the geodesic active contours model by incorporating



(a)

(b)

Figure 2.6 — An example of segmentation using the piecewise-smooth model proposed by Vese-Chan [149]. As a result, the original image is denoised and the contour splits to correctly delineate the three objects captured in the image. (a) Evolution of the segmenting contour superimposed on the original image. (b) Piecewise-smooth image approximation resulting from the segmentation at (a). These images are reproduced from [149].

statistical region-based information. This contributes to overcoming the main weakness of the geodesic active contours, which is their local character, implying convergence to the closest (possibly inconvenient) local minimum and thus strong dependence on initial conditions. Moreover, the model is implemented using the level set method, which eliminates the problems related to topological changes, inherent to the parametric representation of the geodesic active contours. The authors also augmented their model to incorporate texture [113] and motion information [111] in the segmentation.

The general geodesic active regions formulation considers the problem of an image composed of N different regions/phases Ω_i, $i = 1..N$, each delimited by its own contour $\partial\Omega_i$. The proposed energy functional is

$$E(\{\Omega_i\}) = \sum_{i=1}^{N} \left(\alpha \iint_{\Omega_i} r_i(I_0(x,y))\, dx\, dy + (1-\alpha) \int_0^1 g_i(I_0(\partial\Omega_i(p_i))) \left| \frac{\partial\Omega_i}{\partial p_i}(p_i) \right| dp_i \right). \tag{2.41}$$

Here r_i and g_i are two functions which represent the (negative log) likelihood that a particular pixel intensity is observed in region Ω_i, and on the boundary $\partial\Omega_i$, respectively. Moreover, p_i is the parametrization of the curve $\partial\Omega_i$ and $\alpha \in [0,1]$ is a constant weighing the contributions of the two terms. In previous models such as [30, 149], the number of phases was considered as given prior to segmentation and the parameters of the different phases were considered among the unknowns of the variational problem, thus being evaluated and updated during contour evolution. In contrast to these approaches, in the geodesic active regions model the number of phases/regions, as well as the parameters of the likelihoods r_i and g_i, are estimated prior to segmentation from image data * and remain fixed throughout segmentation. This makes the segmentation process more stable and can help avoid unwanted local minima, while introducing dependence on the accuracy of the prior parameter-estimation phase. An example of segmentation using the geodesic active regions models is presented in Fig. 2.7.

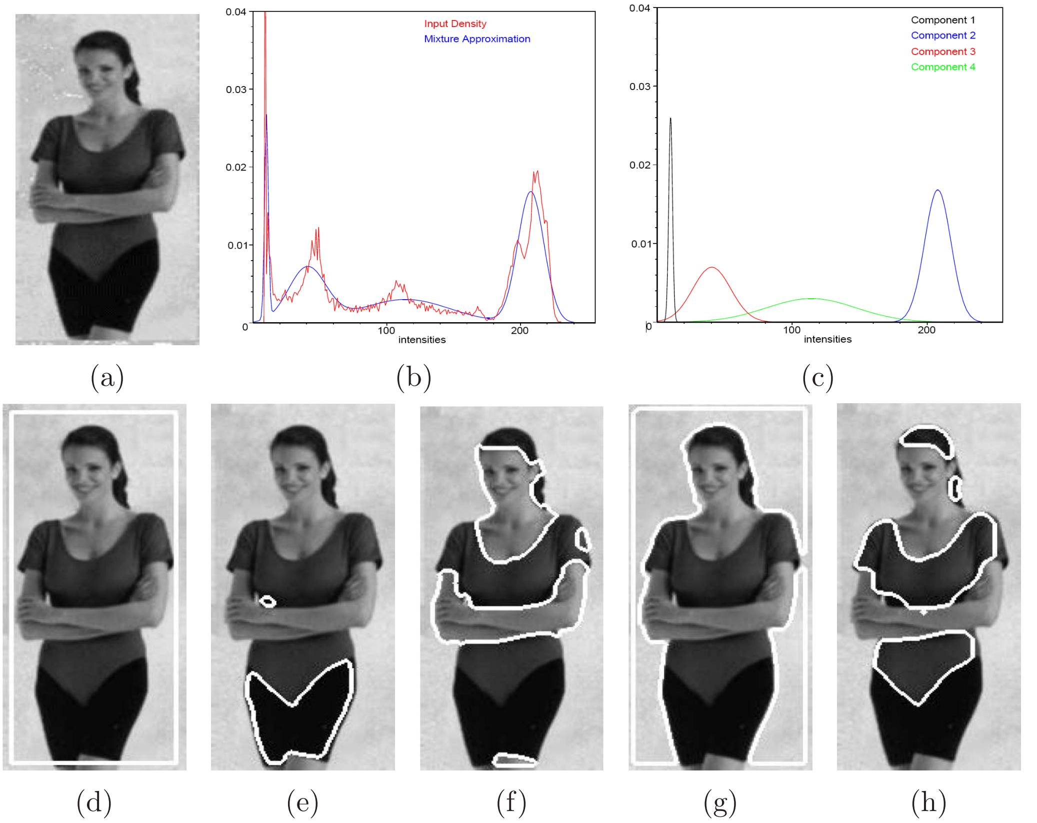


Figure 2.7 — Segmentation using the geodesic active regions model proposed by Paragios [110]. (a) Original image. (b) Image histogram and its approximation with a Gaussian mixture model. (c) The four Gaussian mixture components. (d) Initial contour position. (e)–(g) Final contours corresponding to the four types of regions/components: (e) black pants, (f) skin, (g) background and (h) hair and T-shirt. All images are reproduced from [110].

A further generalization of combined edge- and region-based segmentation approaches is brought by Aubert et al. [10] et Jehan-Besson et al. [80]. They study functionals of the form:

$$E(\Omega_{\text{in}}, \Omega_{\text{out}}, \Gamma) = \iint_{\Omega_{\text{in}}} k^{\text{in}}(x, y, \Omega_{\text{in}})\, dx\, dy + \iint_{\Omega_{\text{out}}} k^{\text{out}}(x, y, \Omega_{\text{out}})\, dx\, dy + \int_{\Gamma} g^{b}(x, y)\, ds. \tag{2.42}$$

---

[*] To obtain the number of regions and the region statistics, the image histogram is modeled as a mixture of Gaussian distributions, each representing one of the regions present in the image. The number of mixture components, as well as their parameters, are then estimated by simultaneously using the "minimum description length" criterion (constraining the number of mixture components) and the maximum likelihood principle (constraining the approximation error of the resulting model). The estimation of boundary probabilities for each pixel is then performed on the basis of the pixel's neighborhood region probabilities.

Here Γ is the contour delineating the object of interest, Ω_{in} and Ω_{out} are the regions lying inside and, respectively, outside the contour Γ, k^{in} and k^{out} are the descriptors of regions Ω_{in} and Ω_{out}, and g^b is the boundary descriptor. The region descriptor is a function which measures the homogeneity of a region. Many of the relevant descriptors are globally attached to the region (region-dependent descriptors), as it is the case with the mean, variance or histogram of a region *. In the case of unsupervised segmentation (i.e. where the region statistics are not learned a priori), these descriptors are updated at each evolution step of the active contour. Their variation during segmentation introduces additional forces into the contour evolution, aspect which has not been considered in previous work. In order to minimize such generalized region-based functionals, the authors introduce a novel approach based on shape-derivation tools [53]. The classical approach for the minimization of combined edge- and region-based functionals was to transform region integrals into boundary integrals, followed by the use of Euler-Lagrange equations and gradient descent to obtain the contour evolution equations (e.g. [110, 166]). This approach becomes complicated in the case of region-dependent descriptors. Aubert et al. [10] show the equivalence of region- and edge-based functionals, which allows the problem to be formulated solely in terms of region functionals, followed by minimization via shape-derivation tools. This facilitates the implementation of new variational region-based formulations, based on region histograms [10] and information-theoretic criteria [76], [21].

The main problem with edge-based segmentation approaches (such as the geodesic active contours), is their susceptibility to become trapped in one of the many local minima in their energy landscape, which implies the sensitivity of the segmentation result to initial conditions. This problem was addressed by Bresson et al. [24], who demonstrate the determination of a global minimum for three proposed active contour models. The models rely on the association of the image segmentation and denoising tasks. All three models are derived from the geodesic active contours, by unification with the total-variation denoising model of Rudin, Osher and Fatemi [127], and with the models developed by Chan and Vese [30, 149] for the approximation of the Mumford-Shah functional in the piecewise-constant and piecewise-smooth case, respectively. Moreover, the authors propose a fast numerical method for the minimization of their models, based on a dual formulation of the minimization. This method avoids the time-consuming operation of re-initializing the level set function to a signed distance function, encountered in most level-set-based variational formulations. An illustration of the denoising and segmentation capabilities of the first model proposed by Bresson et al. is offered in Fig. 2.8.

Summarizing, we have seen that there are several merits of using region-based segmentation approaches over purely edge-based ones. Principally, the introduction of region homogeneity criteria, which are globally defined over the image, renders the segmentation

---

[*] For example, choosing the variance as the descriptor for Ω_{in}, we have:

$$k^{\text{in}}(x, y, \Omega_{\text{in}}) = \frac{\iint_{\Omega_{\text{in}}} (I_0 - \mu_{\text{in}})^2 \, dx \, dy}{\iint_{\Omega_{\text{in}}} dx \, dy},$$

where μ_{in} is the mean intensity of Ω_{in} (also dependent on Ω_{in}).

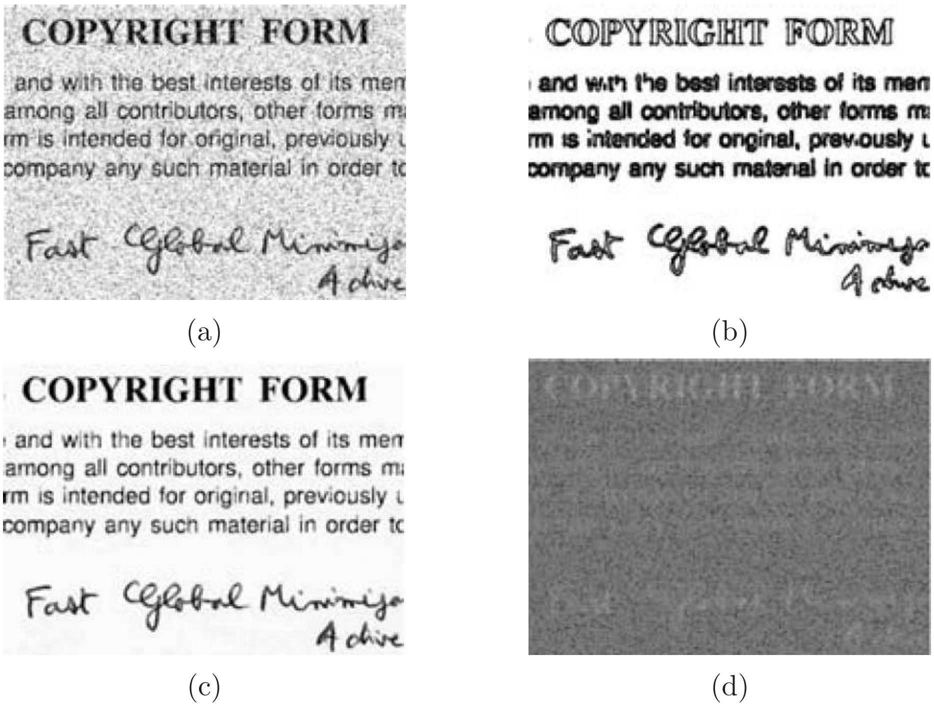

COPYRIGHT FORM
and with the best interests of its men
among all contributors, other forms m
rm is intended for original, previously
company any such material in order tc
Fast Global Minimiza
Active
(a)
(b)
(c)
(d)

Figure 2.8 — Segmentation using the global minimization of the active contour model proposed by Bresson et al. [24]. (a) Original image. (b) Final segmentation contour. (c) Approximated denoised version of the original image. (d) Dual of the approximation at (c). All images are reproduced from [24].

more robust with respect to spurious intensity variations than the locally defined edges. This means that the contour is less susceptible to become trapped into local energy minima, which makes it less sensitive to its initial conditions and capable of converging over larger distances than edge-based contours. Moreover, the region-based segmentation formulations can incorporate a large variety of probabilistic models of the image information in the target regions. This provides an elegant, unitary framework for treating different kinds of features such as image intensity, color, texture or motion.

2.1.4 Introducing Prior Knowledge into Active Contours

All the variational segmentation models presented so far, either edge-based or region-based, rely solely on image information to achieve the segmentation task. However, there are many cases where such information is missing or is corrupted, impairing the attainment of the desired segmentation. Such cases include images with a cluttered background, which do not respect the assumption of homogeneity for the background region, or images where the object of interest is partially occluded by other objects, and thus it cannot be recovered neither based on homogeneity assumptions, nor based on edge information. Finally, other difficult cases are the ones where the image is corrupted by large amounts of noise or blurring, and thus image-based cues are not sufficient to guide the contour towards the desired segmentation. For human perception, such images would cause no problems, since the human visual system would use its a priori knowledge about the world to correct or complete the missing information. Analogously, in computer vision, many variational segmentation approaches deal with such difficult images by introducing a form of a priori knowledge about the objects expected to be found in these images.

Most models use information about the expected shape of object(s) of interest in the image. The classical procedure consists in gathering a set of training shapes of the target object, followed by their alignment with respect to similarity transformations (scale, rotation and translation). Then, the aligned shapes are used to derive a form of statistic knowledge about the target object, which is incorporated into the segmentation model.

Cootes and Taylor [34] are the first to introduce shape knowledge into active contours. They use principal components analysis (PCA) to extract the main modes of variation (principal components) of a set of aligned training shapes represented by points. Then they perform parametric active contour segmentation by restricting contour deformation within the space spanned by the principal components. Their model is shown to cope with cases of missing boundary information and occlusion.

Along the same direction, Leventon et al. [89] perform active contour segmentation in a level set framework, including prior shape information extracted by PCA from a set of aligned training shapes. These training shapes are represented via level set functions, given by the signed distance to their contours. The use of the level set representation eliminates the problem of point management and manual annotation of the training samples to ensure their correspondence, present in [34]. Moreover, the authors motivate their choice by showing that the use of signed distance functions (SDF) provides tolerance with respect to slight misalignments of the training shapes. This is because nearby points of the SDF are highly correlated and thus shape variations are redundantly represented and robustly captured by the PCA.

The segmentation model [89] is not defined by the minimization of a certain energy functional, but rather by the addition of a supplementary term in the evolution of geodesic active contours, which models shape information:

$$\phi(t+1) = \phi(t) + \lambda_1 \left(g\,(c+\kappa)|\nabla\phi(t)| + \nabla\phi(t)\cdot\nabla g\right) + \lambda_2(\phi^*(t) - \phi(t)), \tag{2.43}$$

where λ_1 and λ_2 are two positive constants. We recognize the term weighed by λ_1 as being the geodesic active contour evolution term, which imposes contour smoothness and gradient attraction. The term weighed by λ_2 introduces the shape information via an attraction force towards the prior contour $\phi^*(t)$. This contour is represented in the PCA space (extracted from the training set) and is derived in a separate optimization step, by searching for the maximum a posteriori (MAP) estimates of the position and shape (via PCA coefficients), given the input image and the current contour ϕ. Figure 2.9 presents an example of segmentation using this model in the case of a medical image featuring diffuse object boundaries, which prove to be misleading for purely image-based segmentation.

Tsai et al. [142, 143] follow up on Leventon's model and introduce PCA-derived shape knowledge into region-based segmentation. Their contour is given by a level set function, represented implicitly in terms of a set of shape parameters $\mathbf{w}$ (the PCA coefficients) and

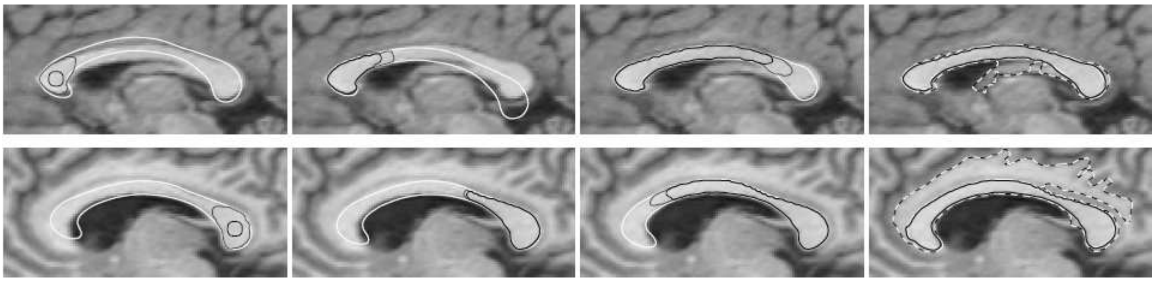
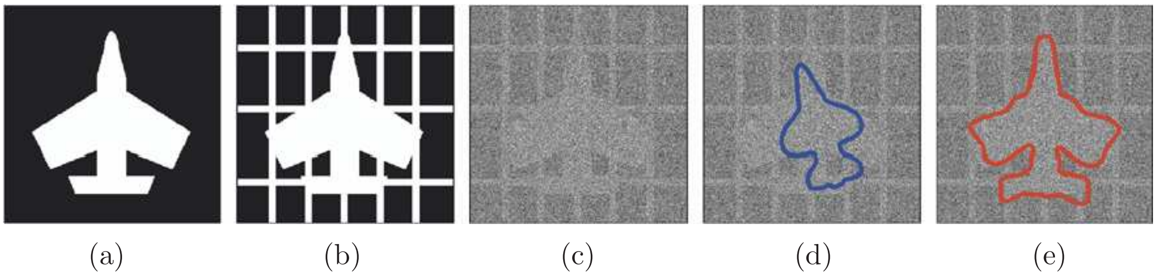


Figure 2.9 — Segmentation of two corpus callosum images using the model of Leventon et al. [89]. The black curve is the segmentation contour at the current evolution step. The gray curve represents the next evolution step. The white contour is the MAP-estimated prior contour. The dotted contour of the final image is the result of standard active contour evolution, which fails to segment the desired structure in the absence of shape influence. All images are reproduced from [89].

pose parameters $\mathbf{p}$ (of a similarity transformation). They propose the following energy functional, which is equivalent (up to a term independent of the contour) with the Chan-Vese piecewise-constant approximation of the Mumford-Shah functional [30]:

$$E(\mathbf{w}, \mathbf{p}) = -\left(\frac{(S^+)^2}{A^+} + \frac{(S^-)^2}{A^-} \right). \tag{2.44}$$

Here $A^\pm = \iint_\Omega H(\pm\phi(\mathbf{w}, \mathbf{p}))\, dx\, dy$ and $S^\pm = \iint_\Omega I_0\, H(\pm\phi(\mathbf{w}, \mathbf{p}))\, dx\, dy$ are the area and intensity sum of the interior and, respectively, exterior region of the zero-level set of $\phi(\mathbf{w}, \mathbf{p})$. The minimization of energy (2.44) with respect to the parameters $\mathbf{w}$ and $\mathbf{p}$ is performed via gradient descent. Along the same lines, the authors proposed two other energy functionals inspired from [159], which segment an image by maximizing the distance between the intensity means and, respectively, variances, of the object and background regions. Figure 2.10 shows an example of segmentation using (2.44), which demonstrates the model's ability to deal with noisy, cluttered images.



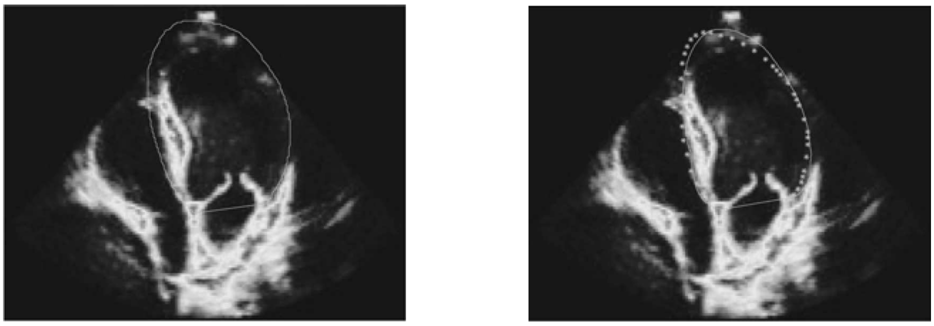
(a) (b) (c) (d) (e)

Figure 2.10 — Segmentation of an airplane image with missing edges using the model of Tsai et al. [143]. (a) Original image. (b) Original image surrounded by line clutter. (c) Image at (b) with additive Gaussian noise. (d) Initializing contour. (e) Final contour. All images are reproduced from [143].

Another approach following the spirit of Leventon's model is the one proposed by Chen et al. [31, 32]. The authors also introduce prior shape information within geodesic active contours, but the nature of this shape information is different: it consists only of the mean of the set of training shapes, whereas Leventon et al. model more complex shape variations via a PCA model. The advantage of Chen's approach consists in the formulation of a unified segmentation model given by an energy functional. This functional is minimized via the calculus of variations and the authors offer a proof of existence for its minimum, which is not feasible in the case of Leventon's two-phase model. The model formulation in terms of parametric contours is

$$E(C, \mu, R, T) = \int_0^1 \left\{ g(|\nabla I_0(C(p))|) + \frac{\lambda}{2} d^2(\mu R C(p) + T, C^*) \right\} |C_p(p)| dp. \tag{2.45}$$

In this equation, $\lambda > 0$ is a fixed parameter, C^* is the mean contour of the training set (i.e. the prior contour), μ, R, T are the parameters of a similarity transformation (scale, rotation and translation) aiming to align the evolving contour C with C^*, and $d((x,y), C^*)$ is the distance from the point (x,y) to contour C^*. The first term of the energy is the classic geodesic active contour and the second one creates the attraction of the (aligned) contour C towards the prior contour C^*. The contour alignment parameters are part of the problem unknowns and are calculated simultaneously with the contour evolution by gradient descent. This parametric contour formulation is extended by the authors to the level set contour representation in a straightforward manner. Figure 2.11 presents an example of segmentation of an epicardium image using Chen's model. As can be seen, the shape prior plays a crucial role in the segmentation, in the absence of well-defined edge or region information.



(a) (b)

Figure 2.11 — Segmentation of an ultrasound image of the epicardium using the model of Chen et al. [32]. (a) Original image with the initial contour. (b) Final segmentation (solid curve) and ground truth outlined by the expert (dotted line). All images are reproduced from [32].

The use of a new shape model in conjugation with the geodesic active regions framework [112] was proposed by Rousson and Paragios in [126]. Their shape model is formulated in terms of the level set representation and accounts both for a global shape ϕ_M, estimated as the mean shape of a training set, and for local shape variability, given by the local variance $\sigma_M(x,y)$. Formally, shape knowledge is encapsulated as a pixel-wise Gaussian probability model of the level set function:

$$p^M_{(x,y)}(\phi) = \frac{1}{\sqrt{2\pi}\sigma_M((x,y))} e^{-\frac{(\phi(x,y) - \phi_M(x,y))^2}{2\sigma_M^2(x,y)}}. \tag{2.46}$$

The model parameters ϕ_M and σ_M are estimated by maximizing the model likelihood with respect to the training data, under a smoothness constraint for σ_M and while performing periodic reinitialization of ϕ_M to a signed distance function. The resulting shape prior is then introduced into the geodesic active regions model [112] as a supplementary energy term of the form:

$$\begin{aligned} E(\phi, A) &= -\iint_\Omega H_\epsilon(\phi(x,y)) \log\left(p^M_{A(x,y)}(s\phi(x,y))\right) dx\,dy \\ &= \iint_\Omega H_\epsilon(\phi(x,y)) \left(\log \sigma_M(A(x,y)) + \frac{(s\phi(x,y) - \phi_M(A(x,y)))^2}{2\sigma_M^2(A(x,y))}\right) dx\,dy. \end{aligned} \tag{2.47}$$

Here $A = (s, a_1, a_2, \dots a_N)$ are the parameters of a linear transformation, out of which s is the scale parameter. The introduction of the Heaviside function H_ϵ restricts integration to the inside of the region of interest, thus making level set function comparison independent with respect to the size of the domain Ω. The minimization of energy (2.47) amounts to maximizing the likelihood of the aligned level set function $\phi(A(x,y))$, given the prior model, under the assumption that the probability densities (2.46) are independent across pixels. The optimization with respect to the alignment parameters A is performed through gradient descent. An example of segmentation using energy (2.47) in association with the geodesic active region model is presented in Fig. 2.12.

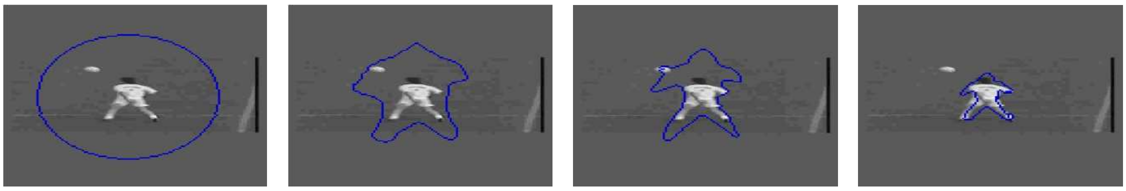


Figure 2.12 — Segmentation of a low-quality image of a football player using the model of Rousson and Paragios [126]. The shape prior helps capture the whole silhouette of the player, despite it being composed of regions of different gray-value (head, body, legs). All images are reproduced from [126].

The integration of a statistical shape prior into the Mumford-Shah functional and its cartoon limit was proposed by Cremers et al. in [39, 43]. They use a parametric contour representation based on B-spline curves and introduce a shape model given by the Gaussian distribution of the shape parameters, which are the spline control points:

$$P(C) \propto \exp\left(-\frac{1}{2}(C-\mu)^{\mathrm{T}}\Sigma_\perp^{-1}(C-\mu)\right). \tag{2.48}$$

Here $C = (x_1, y_1, \dots x_N, y_N)^\top$ is the vector of control points and μ, $\sigma_\perp$ are the mean and, respectively, regularized covariance matrix, estimated from the aligned control-point vectors of a set of training contours representing the object of interest. The proposed energy functional combines an image-based energy derived from the Mumford-Shah functional with a term corresponding to the prior shape information:

$$
\begin{aligned}
E(I,C) &= E_{\text{image}}(I,C) + \alpha\, E_{\text{prior}}(C),\\
E_{\text{image}}(I,C) &= \frac{1}{2}\iint_\Omega (I-I_0)^2\,dx\,dy + \frac{\lambda^2}{2}\iint_\Omega w_C(x,y)|\nabla I|^2\,dx\,dy + \nu\int_0^1 C_s^2(s)ds, \qquad (2.49)\\
E_{\text{prior}}(C) &= -\log P(C) = -\frac{1}{2}(C-\mu)^{\mathrm{T}}\Sigma_\perp^{-1}(C-\mu).
\end{aligned}
$$

Here I is the piecewise-smooth approximation of the original image I, w_C is a contour indicator function such that $w_C(x,y) = 0$ if $(x,y) \in C$ and $w_C(x,y) = 1$ otherwise. The last term of $E_{\text{image}}(I,C)$ is a modification of the classical length term $L(C) = \int_0^1 |C_s(s)|$, which prevents the control points from clustering together and causing numerical instability problems, as is the case with typical parametric active contours. Furthermore, the authors render their energy functional invariant with respect to similarity transformations (scaling, rotation and translation), by replacing the control-point vector C with an analytic expression which aligns it with respect to the mean control-point vector μ. This bypasses the need for optimization with respect to the alignment parameters, which can fail to converge to the desired minimum. Figure 2.13 shows a segmentation example which demonstrates the model's ability to cope with clutter, unlike its variant without shape prior or the classical geodesic active contour.

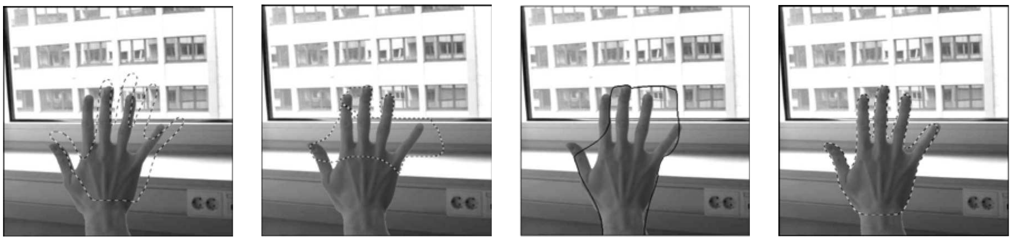
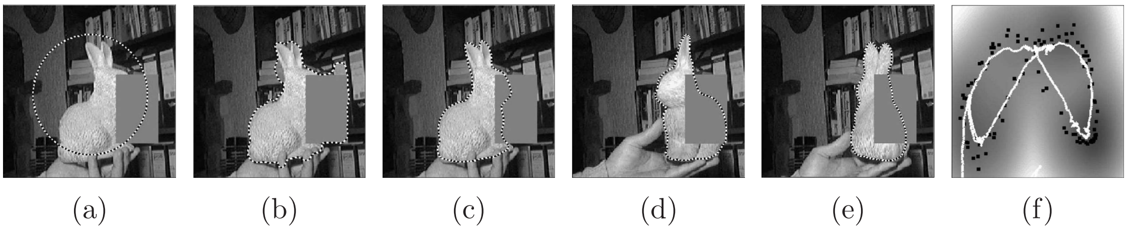

(a) (b) (c) (d)

Figure 2.13 — Segmentation of an image affected by clutter using the model of Cremers et al. [43]. (a) Initial contour on cluttered image. (b) Segmentation using model (2.49), without the shape prior term. (c) Segmentation using geodesic active contours. (d) Segmentation using prior shape information (2.49). All images are reproduced from [43].

The presented approaches for introducing prior shape information into the segmentation framework used the assumption of a Gaussian distribution of the permissible shapes. This means that any such shape can only be represented as a linear combination of a set of eigenmodes, such as the ones obtained by PCA. This assumption is quite limiting in many realistic situations where the shapes undergo more complex transformations. For instance, different 2D views of a 3D object will not belong to a Gaussian distribution. In this context, Cremers et al. [42, 44] extend their formulation [39, 43] to a nonlinear shape model. They use a novel method of density estimation which can be considered as an extension of kernel-PCA to a probabilistic framework. In particular, the authors apply a nonlinear mapping to the training data, resulting in a higher-dimensional representation that is considered to be Gaussian-distributed. This makes the distribution in the original space highly non-Gaussian and permits the encoding of complex shape deformations. Figure 2.14 presents such an example, where the shape prior models different views of a rotating rabbit and allows its accurate tracking through clutter and occlusion.



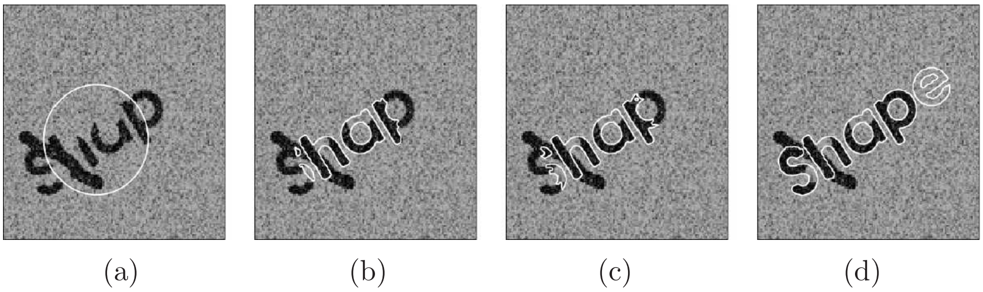
(a) (b) (c) (d) (e) (f)

Figure 2.14 — Tracking of a rotating rabbit through clutter and occlusion, using the model of Cremers et al. [42, 44]. (a) Initial contour. (b) Segmentation without prior. (c)–(e) Segmentation of different views with nonlinear prior. (f) Training data (black dots), estimated energy density and contour evolution (white curve) in appropriate 2D projections (onto 1st and 2nd principal components). Evolution is following the valleys of low energy created by the training data. All images are reproduced from [44].

In the context of integrating prior shape knowledge into level set based segmentation, Cremers and Soatto [41] investigate dissimilarity measures for shapes encoded by the signed distance function. They advocate the use of symmetry in the construction of dissimilarity measures. Furthermore, they propose a new dissimilarity measure, which is symmetric, not biased towards small areas and constitutes a pseudo-distance (since it does not satisfy the triangle inequality). Given two shapes ϕ_1, ϕ_2, represented via the signed distance function, this measure is defined as

$$d^2(\phi_1, \phi_2) = \iint_\Omega (\phi_1 - \phi_2)^2 \, \frac{h(\phi_1) + h(\phi_2)}{2} \, dx\, dy. \tag{2.50}$$

The use of the normalized Heaviside function $h(\phi) = \dfrac{H(\phi)}{\iint_\Omega H(\phi)\, dx\, dy}$ prevents the bias towards small areas. Figure 2.15 offers a comparison of using this measure and its asymmetric or un-normalized versions, for encoding the attraction towards the shape prior in the context of segmentation with the piecewise-constant Chan-Vese model.



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

Figure 2.15 — Comparison of different dissimilarity measures. (a) Initial contour. (b)–(d) Segmentation using a prior which encodes the entire word “shape”, based on an un-normalized and asymmetric dissimilarity measure — (b), its normalized version — (c) and the symmetric normalized measure given by (2.50) — (d). Neither of the two asymmetric measures is able to propagate the shape prior outside the initial interior shape area. All images are reproduced from [41].

Another approach for introducing shape prior information into level-set-based segmentation was presented by Cremers et al. in [45, 48]. The main innovation is the introduction of a multi-modal statistical shape prior, which allows the encoding of multiple, fairly different training shapes (see Fig. 2.16). This prior is obtained by applying the classical Parzen-Rosenblatt density estimator [116, 124] to the level set representation:

$$P(\phi) \propto \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{1}{2\sigma^2} d^2(H\phi, H\phi_i)\right). \tag{2.51}$$

Here $\{\phi_i\}_{i=1..N}$ is the set of training shapes, $H\phi$ is the Heaviside function applied to the level set function ϕ, σ is the kernel width of the Parzen-Rosenblatt estimator and the distance d is given by:

$$d^2(H\phi_1, H\phi_2) = \iint_\Omega \left(H(\phi_1(x,y)) - H(\phi_2(x,y))\right)^2 dx\, dy. \tag{2.52}$$

The authors render their distance function invariant to shape translation by evaluating the level set function in relative coordinates with respect to its gravity center. Then they introduce the shape prior into a segmentation scheme based on the Chan-Vese piecewise-constant model:

$$E(\phi) = \frac{1}{\alpha} E_{\text{CV}}(\phi) + E_{\text{shape}}(\phi), \tag{2.53}$$

where E_{CV} is the Chan-Vese energy (2.33) and $E_{\text{shape}} = -\log P(\phi)$ is the shape prior energy. Using (2.51) the authors encode various poses of a walking person (Fig. 2.16, first row). The resulting shape prior enables them to successfully track a partially occluded person, as can be seen in Fig. 2.16, second row.

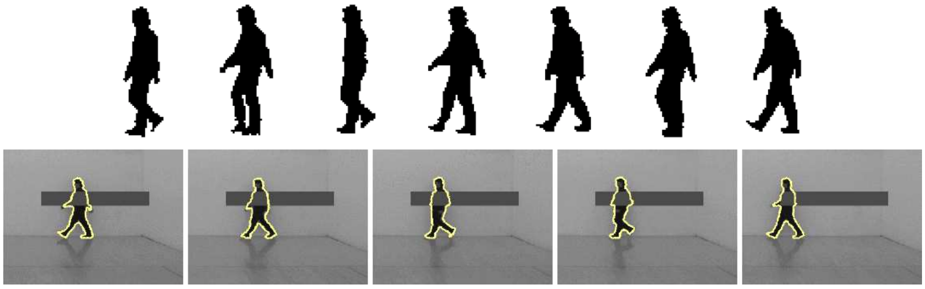


Figure 2.16 — Segmentation of a partially occluded walking person with the model of Cremers et al. [45, 48]. First row: sample training shapes. Second row: Examples of segmented frames using the proposed shape prior model. All images are reproduced from [45].

A unified segmentation framework which integrates edge- and region-based information with a geometric shape prior was proposed by Bresson et al. in [21, 23]. The proposed model consists of an energy functional composed of three complementary terms:

$$\begin{aligned} E(C, \mathbf{x}_{\text{pca}}, \mathbf{x}_T, I_{\text{in}}, I_{\text{out}}) = {} & \beta_s\, E_{\text{shape}}(C, \mathbf{x}_{\text{pca}}, \mathbf{x}_T) + \beta_b\, E_{\text{boundary}}(C) \\ & + \beta_r\, E_{\text{region}}(\mathbf{x}_{\text{pca}}, \mathbf{x}_T, I_{\text{in}}, I_{\text{out}}). \end{aligned} \tag{2.54}$$

The prior shape information is obtained similarly to Leventon et al. [89], by performing PCA on a set of signed-distance functions to the aligned training shape contours. This serves the representation of a prior shape function, embedding the prior shape contour, in terms of the coefficients $\mathbf{x}_{\text{pca}}$ of its projection onto the PCA eigenvectors. This prior shape function is close to a signed-distance function and is given by:

$$\hat{\phi} = \overline{\phi} + \mathbf{W}_p \, \mathbf{x}_{\text{pca}}, \tag{2.55}$$

where $\overline{\phi}$ stands for the mean of the training data and $\mathbf{W}_p$ is the matrix of the PCA eigenvectors. The vector $\mathbf{x}_T$ from (2.54) contains the parameters of a geometric (similarity or affine) transformation, acting on the planar coordinates (x, y) of the prior contour. During the minimization of (2.54), the prior shape function is updated at the same time as the contour C, by gradient descent equations for the parameters $\mathbf{x}_{\text{pca}}$ and $\mathbf{x}_T$.

The first term of the energy (2.54) is a functional introduced by Bresson et al. in [22], which evaluates the shape difference between the contour C and the zero level set of the prior shape function $\hat{\phi}$ provided by the PCA:

$$E_{\text{shape}}(C, \mathbf{x}_{\text{pca}}, \mathbf{x}_T) = \oint_0^1 \hat{\phi}^2(\mathbf{x}_{\text{pca}}, h_{\mathbf{x}_T}(C(q))) \, |C_q(q)| \, dq. \tag{2.56}$$

We recognize here the classic contour length term $\oint_0^1 |C_q(q)| dq$, where the integration along the contour is weighed at each point by the factor $\hat{\phi}^2(\mathbf{x}_{\text{pca}}, h_{\mathbf{x}_T}(C(q)))$. Since $\hat{\phi}$ approximates a signed-distance function, this factor approximates the shortest distance between current integration point $C(q)$ and the prior contour, given by the zero level set of $\hat{\phi}$. The transformation $h_{\mathbf{x}_T}$ is meant to align contour C with the prior contour. The second term of the energy (2.54) is the edge-attraction term of the geodesic active contours model, which allows the segmentation to capture local structure variations:

$$E_{\text{boundary}}(C) = \oint_0^1 g(|\nabla I_0(C(q))|) \, |C_q(q)| \, dq. \tag{2.57}$$

The third term of the energy (2.54) drives the shape prior $\hat{\phi}$ globally towards a homogenous intensity region, via the piecewise-smooth Mumford-Shah functional:

$$\begin{aligned} E_{\text{region}}(\mathbf{x}_{\text{pca}}, \mathbf{x}_T, I_{\text{in}}, I_{\text{out}}) = & \iint_\Omega ((I_0 - I_{\text{in}})^2 + \mu |\nabla I_{\text{in}}|^2) \, H(\hat{\phi}(\mathbf{x}_{\text{pca}}, \mathbf{x}_T)) \, dx \, dy \\ & + \iint_\Omega ((I_0 - I_{\text{out}})^2 + \mu |\nabla I_{\text{out}}|^2) \, H(-\hat{\phi}(\mathbf{x}_{\text{pca}}, \mathbf{x}_T)) \, dx \, dy. \end{aligned} \tag{2.58}$$

Here I_{in} and I_{out} designate the piecewise-smooth approximations of the original image I_0 inside and, respectively, outside the zero level set of the prior shape function $\hat{\phi}$ and H is the Heaviside function. The authors also provide a level set representation of their model and offer a proof of existence of its minimizer. Figure 2.17 shows an example of medical image segmentation, outlining the merit of the model in accurately guiding the segmenting contour towards the object of interest, despite locally misleading information.

The simultaneous segmentation of multiple familiar objects, using multiple competing shape priors, was treated by Cremers et al. in [46, 49]. They introduce a labeling function,

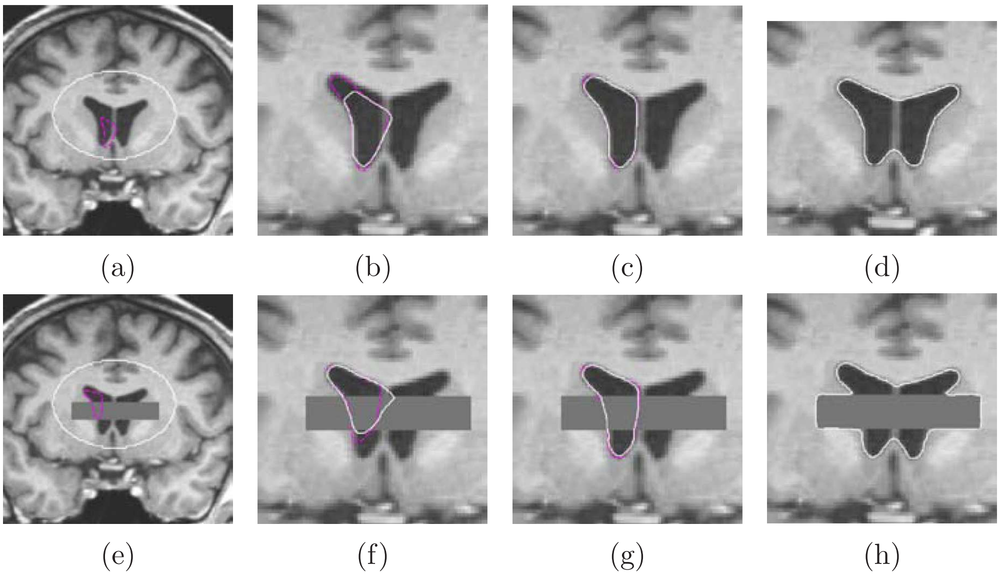


Figure 2.17 — Segmentation of the left brain ventricle in a MRI image with the method proposed by Bresson et al. [23]. First row: original image segmentation. Second row: segmentation of the original image featuring an occlusion. (a)–(c), (e)–(g) Segmentation with the model of Bresson et al. [23]: contour as the white curve, prior contour as the magenta curve. (d),(h) Segmentations using geodesic active contours [28] and, respectively, the piecewise-smooth model of Vese and Chan [149], both failing to capture the object of interest. All images are reproduced from [23].

defined over the image domain, which indicates where to apply certain priors. By optimizing this function simultaneously with the level set function, the authors jointly generate a segmentation and a partition of the image domain among the objects of interest. For the case of two competing priors, embedded in level set functions ϕ_1 and ϕ_2, the associated energy is:

$$\begin{aligned} E_{\text{shape}}(\phi, L) = &\iint_\Omega \frac{(\phi-\phi_1)^2}{\sigma_1^2}(L+1)^2\,dx\,dy + \iint_\Omega \frac{(\phi-\phi_2)^2}{\sigma_2^2}(L-1)^2\,dx\,dy \\ &+ \gamma \iint_\Omega |\nabla L|\,dx\,dy. \end{aligned} \tag{2.59}$$

Here $\sigma_i^2 = \iint_\Omega \phi_i^2\,dx\,dy - (\iint_\Omega \phi_i\,dx\,dy)^2$ represents the variance of ϕ_i and $L : \Omega \to \mathbb{R}$ is the labeling function, which enforces the prior which is most similar to the level set ϕ at each image location. For fixed ϕ, the first two terms of the energy induce the following qualitative behavior of the labeling: $L \to 1$ if $|\phi - \phi_1|/\sigma_1 < |\phi - \phi_2|/\sigma_2$ and $L \to -1$ if $|\phi - \phi_1|/\sigma_1 > |\phi - \phi_2|/\sigma_2$. Moreover, in (2.59), $\gamma > 0$ and the last term imposes smoothness of the labeling function. The authors extend this formulation to multiple priors, by considering a vector-valued labeling function $\mathbf{L} : \Omega \to \mathbb{R}^n$, $\mathbf{L}(x, y) = (L_1(x, y), \ldots L_n(x, y))$. Using this function, the authors employ the $m = 2^n$ vertices of the polytope $[-1, 1]^n$ to encode m different regions, denoted by their respective indicator functions χ_i, $i = 1..m$ (depending on the vector $\mathbf{L}$). This results into an energy of the form

$$\begin{aligned} E_{\text{shape}}(\phi, \mathbf{L}) = \sum_{i=1}^{m-1} \iint_{\Omega} \frac{(\phi - \phi_i)^2}{\sigma_i^2} \chi_i(\mathbf{L})\, dx\, dy + \lambda^2 \iint_{\Omega} \chi_m(\mathbf{L})\, dx\, dy \\ + \gamma \sum_{i=1}^{n} \iint_{\Omega} |\nabla L_i|\, dx\, dy. \end{aligned} \tag{2.60}$$

Here $\lambda > 0$ and the second energy term corresponds to a region where no prior is imposed, since the resemblance between ϕ and any of the priors falls bellow a threshold dictated by λ. This allows pure image-based segmentation of objects for which no prior is available. An example of segmentation of three familiar objects with missing parts and of a fourth unknown object (in the center) is presented in Fig. 2.18. The results were obtained by combining the shape prior energy (2.60) with the piecewise-constant Chan-Vese model, as in (2.53).

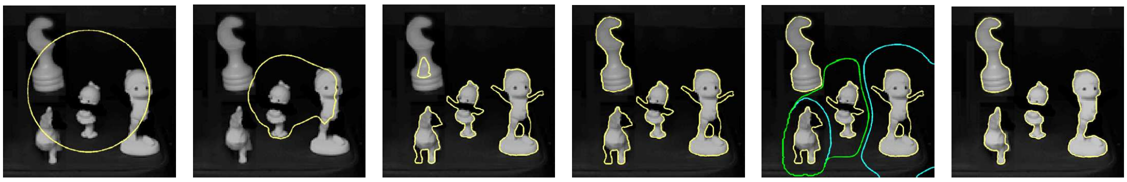

(a) (b) (c) (d) (e) (f)

Figure 2.18 — Segmentation of multiple familiar objects using multiple competing shape priors, using the model of Cremers et al. [45, 49]. (a) Initial contour. (b)–(d) Segmentation with multiple competing priors. (e) Zero-level contours of the two labeling functions which designate the influence regions of the priors (blue and green curves). (f) Segmentation without shape priors. All images are reproduced from [45].

2.1.5 Object Tracking with Variational Segmentation Methods

Variational segmentation methods can also be applied to perform object tracking through image sequences. The goal of object tracking is to recover the position and deformation of the object throughout the image sequence. To this end, a suitable approach is to follow the object contour evolution within the image sequence, task for which, as we have seen, a wealth of versatile variational methods have been developed.

In this context, Kass et al. [81] directly apply their original snakes formulation to track a person's lips. The approach consists of simply segmenting each image in turn, using as initial contour the final segmentation contour of the previous image. This method is successful within the known limitations of the snakes approach and given that the motion is slow with respect to the frame rate. Similarly, Cremers et al. [44, 48] report tracking by direct frame-by-frame segmentation, where segmentation is performed via more sophisticated models that include statistical shape priors. However, the use of static shape prior models (i.e., constant throughout the image sequence) restricts the possible tracked shape deformations to the probabilistic distributions which are considered. The consistency between consecutive frames is exploited by Paragios and Deriche in [111]. They incorporate the typical assumption used in tracking scenarios — the constant brightness of the target object [78, 92] — within a variational framework which includes a boundary-attraction term and a background-substraction probabilistic term. This enables them to perform simultaneous tracking and motion estimation (by the recovery of optical flow vectors). The same constant brightness assumption is integrated into a region-based segmentation energy used for tracking in [97]. Based on the same constant brightness principle, Yilmaz et al. [163] propose an energy functional combining region-based color and texture cues with an online shape model, learned from contour deformation during the course of tracking.

More elaborated tracking techniques incorporate prediction mechanisms such as Kalman filtering [119, 141] or particle filtering [121, 122]. Generally, prediction relies on learning a model of object displacement based on prior observations. Such a model is used to provide an initial guess of the object position in a subsequent frame, which is further refined based on image data. The final position of the detected object is then used to improve the prediction model.

Addressing the limitations of tracking by segmentation with static prior shape models, Cremers [40] proposes the use of dynamical statistical shape priors for level-set-based segmentation and tracking. The author distinguishes shape, represented by the embedding function ϕ, from shape transformation T_θ, which acts on the grid and implicitly transforms the shape into $\phi(T_\theta(x, y))$ (e.g. similarity transformation). Considering a set of consecutive images from an image sequence $I_{1:t} = \{I_1, \ldots I_t\}$, Cremers formulates the segmentation of I_t as a problem of Bayesian inference, where one wants to maximize the conditional probability:

$$P(\phi_t, \theta_t | I_{1:t}) = \frac{P(I_t|\phi_t, \theta_t, I_{1:t-1})P(\phi_t, \theta_t|I_{1:t-1})}{P(I_t|I_{1:t-1})} \tag{2.61}$$

with respect to the embedding function ϕ_t and the transformation parameters θ_t. After a few assumptions, the problem is simplified to the maximization of:

$$P(\phi_t, \theta_t|I_t, \hat{\phi}_{1:t-1}, \hat{\theta}_{1:t-1}) \propto P(I_t|\phi_t, \theta_t)P(\phi_t, \theta_t|\hat{\phi}_{1:t-1}, \hat{\theta}_{1:t-1}) \tag{2.62}$$

with respect to ϕ_t and θ_t, where $\hat{\phi}_{1:t-1}$ and $\hat{\theta}_{1:t-1}$ are the estimated segmentations and transformations for the previous images $I_{1:t-1}$. Towards reliable statistical estimation, data dimensionality is reduced by PCA, which results in the level set function ϕ being represented by its projection $\boldsymbol{\alpha} \in \mathbb{R}^n$ onto the set of principal components, learned from training data. Assuming independence of shape and transformation parameters, as well as a uniform distribution of the transformation parameters, the estimation of the second factor of (2.62) reduces to the estimation of the conditional probability $P(\boldsymbol{\alpha}_t|\hat{\boldsymbol{\alpha}}_{1:t-1})$, which encapsulates the dynamics of the shape deformation. The author models this probability by an auto-regressive model of order k:

$$P(\boldsymbol{\alpha}_t|\hat{\boldsymbol{\alpha}}_{1:t-1}) \propto \exp\left(-\frac{1}{2}\mathbf{v}^\top\Sigma^{-1}\mathbf{v}\right), \tag{2.63}$$

where

$$\mathbf{v} = \boldsymbol{\alpha}_t - \boldsymbol{\mu} - A_1\hat{\boldsymbol{\alpha}}_1 - A_2\hat{\boldsymbol{\alpha}}_2 \ldots A_k\hat{\boldsymbol{\alpha}}_k. \tag{2.64}$$

and the model parameters, given by the mean $\boldsymbol{\mu} \in \mathbb{R}^n$, and the transition and noise matrices $A_1, \ldots A_k$, $\Sigma \in \mathbb{R}^{n \times n}$, are estimated from training data. The author also considers a joint dynamic model of shape and transformation parameters, based on the same kind of autoregressive model and using a concatenated vector of shape parameters and transformation-parameter differences $\tilde{\boldsymbol{\alpha}}_t = (\boldsymbol{\alpha}_t, \triangle\theta_t)^\top$. The maximization of (2.62) is equivalent to minimizing the negative logarithm of (2.62), which can be translated into the variational optimization of an energy of the form:

$$E(\boldsymbol{\alpha}_t, \theta_t) = E_{\text{data}}(\boldsymbol{\alpha}_t, \theta_t) + \nu E_{\text{shape}}(\boldsymbol{\alpha}_t, \theta_t). \tag{2.65}$$

The author models $E_{\text{data}}(\boldsymbol{\alpha}_t, \theta_t)$ by assuming pixel-wise Gaussian distributions for the object of interest and the background and $E_{\text{shape}}(\boldsymbol{\alpha}_t, \theta_t)$ is given by the negative logarithm of (2.63) (or its variant for the joint modeling of shape and transformation parameters). Manually segmented sequences of a walking person are used to estimate the parameters of the autoregressive model. Then, the segmentation of noisy sequences of the walking person is performed using the proposed energy (2.65). Results of this method are presented in Fig. 2.19.

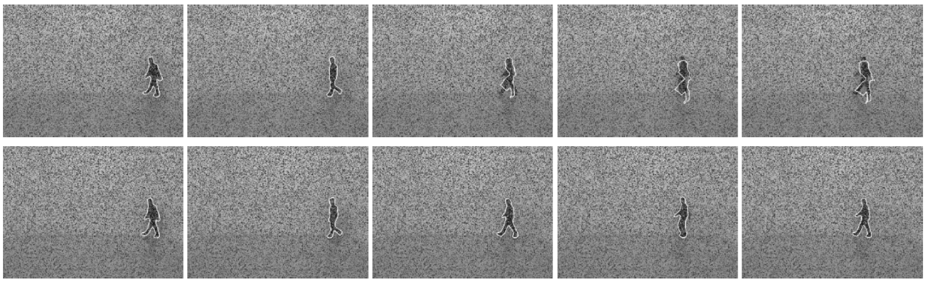


Figure 2.19 — Segmentation of a noisy sequence of a walking person. First row: segmentation using a static prior, given by a uniform probability in the space of the first few eigenmodes. The process is stuck in a local minimum after the first frames. Second row: segmentation using the dynamic prior (energy (2.65)) proposed by [40]. The dynamic prior helps the segmentation cope with the misleading image information due to noise. All images are reproduced from [40].

2.2 Behavior Recognition with Computer Vision Approaches

Behavior recognition is currently a very active research field, with the majority of efforts dedicated to human behavior modeling and recognition. Good reviews of the topic can be found in [2, 64]. A comprehensive survey of the literature in the field is performed by Moeslund et al. in [101], which reviews a number of 424 publications in major conferences and journals, 352 of which are recent, i.e. published between years 2000 and 2006. On the one hand, the great popularity of the field can explained by the scientific challenge posed by many behavior recognition problem, involving complex, ill-posed problems, such the motion estimation and understanding of a self-occluding, non-rigid 3D object from 2D images. On the other hand, the increased interest in behavior recognition is given by the wealth of its applications, including automated surveillance, human-computer interaction, medical diagnosis, sports performance analysis and video indexing and retrieval.

The general trend reflected in the behavior recognition approaches that we have found in the literature is the decomposition of the resolution in two sequential processes. The first one is a feature extraction process, where features considered relevant for the recognition task are extracted from the image sequence. The second one is the actual behavior recognition process, which generally solves a classification problem in terms of the extracted features. The main difference of our framework with respect to these approaches is the joining of the two processes — feature extraction and behavior recognition. The purpose of our approach is to enable the two processes to collaborate towards improved results for both, as will be shown in Chapter 3. In the following, we will start with a few considerations relative to the definition of the behavior recognition task. Then, we will present some of the approaches for behavior recognition encountered in the literature. Our focus will be on emphasizing the separation of the feature extraction and behavior recognition processes. Moreover, we will outline the methods employed for each of these processes, and relate them to the methods used in our framework.

The task of behavior recognition can be approached at different levels of detail, depending on the particular application which is considered and on the complexity involved in the targeted behavior. In order to allow for a more clear delimitation of research goals in this direction, several behavior hierarchies have been proposed, involving different terms, such as action, activity, complex action, etc. The early work of Nagel [105] suggests the use of a hierarchy composed of “change, event, verb, episode and history”. More recently, Bobick [16] proposed the use of a hierarchy consisting of “movement, action and activity”, whereas Moeslund et al. [101] base their survey on a hierarchy composed of “action primitives, actions and activities”. In the latter work, actions are made up of atomic units which are the action primitives. Furthermore, activities are larger scale events, composed of actions, and potentially involving causal relations, interactions among humans or with objects in the environment. In our thesis, we denote the atomic primitives as “actions”, and define “behavior” as a sequence of actions. Our general framework does not specifically focus on human behavior, but rather uses the generic concept of object, which includes humans, (moving) machines or animals. We delimit the scope of our work to single object behavior recognition and do not consider the higher level of interactions between objects (or humans), as specified by the “activity” category in the hierarchy of [101].

Another aspect which differentiates the multitude of existing approaches for behavior recognition is the level of detail in the modeling and tracking of objects, which depends on the targeted recognition application. At the coarsest level, the objects are represented by their centroids or by their bounding boxes or ellipses. Such representations serve applications where the recognition and understanding of behavior can be performed in terms of the moving trajectories of these centroids / bounding boxes. For instance, the recognition of two-person pedestrian interactions is considered by Sato and Aggarwal [131], who track persons as moving boxes and classify the motion patterns of the boxes. On a more detailed analysis level, the silhouettes / contours of the human body as a whole are extracted from the input images. For example, Yu et al. [164] extract human contours and their PCA representation from image sequences. Then they input the resulting trajectories in eigenspace into a neural network to discriminate among the actions of "walking", "running" and "other".

More precise modeling is obtained by distinguishing (in 2D) individual human body parts, such as the head, torso, arms and legs. An example in this respect is Wren et al.'s tracking system [157], which yields a human body representation in terms of 2D blobs, associated with the head, torso, hands and feet. The system is used, among others, for the gesture control of two virtual reality applications. In similar vein, Starner and Pentland [139] track the human hands as 2D blobs and use HMMs to recognize a subset of the American Sign Language (ASL). For applications concerning hand-gesture recognition, more detailed descriptions of the hand need to be employed. For instance, Lockton and Fitzgibbon [91] extract the hand mask based on skin color and recognize finger-spelling by using a nearest-neighbor classification technique. Birk et al. [13] extract a PCA representation of the hand image on a dark background (normalized with respect to similarity transformations) and recognize finger-spelling at each frame by maximum likelihood estimation in the space of the PCA coefficients.

The most complex descriptions of the human body (or of its parts), employed for motion modeling, are 3D volumetric models. To our knowledge, complete systems which perform 3D motion reconstruction and behavior recognition have not been developed so far. However, there is a large body of work regarding 3D tracking and motion reconstruction at various complexity levels, either using stereo information acquired with multiple cameras or using monocular image sequences, together with constraints on kinematics and movement (e.g. [52, 137, 147]). Evidently, the results of these approaches can potentially be used in behavior recognition tasks. Moreover, there are several behavior recognition approaches which circumvent the vision problem and assume the existence of information regarding body posture, generally in terms of joint locations, either in 2D or in 3D (e.g. [114, 151, 161]).

Finally, another category of approaches do not explicitly employ a method for object modeling or tracking, but rather contain an implicit description of the object(s) of interest, by modeling the regions of motion within the image sequence. Since they do not attempt to explicitly identify the object(s) of interest, these methods generally assume that all moving regions correspond to such objects (and are thus perturbed by regions not obeying this rule). An example of such a motion-only approach is the method proposed by Bobick and Davis [17]. They model the motion within an image sequence by extracting motion energy images (MEI), which indicate the presence / absence of motion at a certain pixel, and motion history images (MHI), where pixel intensities are functions of the recency of motion at a certain pixel. From the MEIs and MHIs, they extract Hu moments, that they subsequently use to classify the image sequence in terms of the shortest Mahalanobis distance to learned models of each action. In similar spirit to the MHIs, Yi et al. [160] extract motion characteristics of an image sequence via pixel change ratio maps (PCRM). Then, they compute a motion histogram from the PCRM, which is used for classification in terms of the Euclidian distance with respect to the histograms of training sequences.

A sensitive point of these approaches is the ambiguity among different motions, induced by the integration of information throughout the whole image sequence. Another inherent difficulty is posed by the imprecise nature of the motion detection strategy (e.g., inner regions of moving objects may not detected).

In our general framework, the level of detail that we use for image analysis and object modeling is the highest permitted by a 2D representation and by the fact that we do make any assumptions regarding the object targeted for behavior recognition (i.e., we do not specify a particular 2D or 3D object model). That is, we represent the target object by its contour within the image. This allows us to extract any object attributes which are functions of the contour and the image, such as the color, texture properties or position (in 2D). Obviously, such a representation is more general than blob or bounding box representations, since the latter can be extracted from it.

Regarding the strategy used for recognition, we notice that according to the targeted application, some of the existing approaches perform a frame-by-frame classification of the input image sequences. The features extracted from each image are classified into one of the given action classes, using the information gathered from training data, via methods like maximum likelihood ([13]) or nearest-neighbor template matching, to which a deterministic boosting method is added for speed-up ([91]). These methods are limited to cases where a static recognition of each frame is feasible, without the need of using context information from adjacent frames.

Alternatively, other approaches use features extracted from the whole image sequence to globally classify it as one of a set of possible actions. In this direction, [17] and [160] use nearest-neighbors methods for classification. Moreover, Efros et al. [58] track humans in image sequences and eliminate global motion by extracting a window centered on the tracked person. From the sequence within this window, they further compute a set of motion features based on the blurred optical flow, which capture the residual motion of the person's body parts. Then, the features are matched with a database of learned motions using spatio-temporal cross-correlation. One difficulty faced by this kind of methods arises due to the differences in the speed of performing the compared actions. This creates the necessity for temporal alignment between the compared sequences (when the feature set dimension scales with the dimension of the image sequence, e.g. [58]), or for adjustment of the temporal parameters used in computing global features over the image sequence (e.g. [17]).

Another approach leading to the global classification of a sequence into one of a set of (atomic) actions is that of extracting "space-time shapes" from the XYT volume of the image sequence. Yilmaz and Shah [162] construct a spatio-temporal volume (STV) by considering the 2D contour of a person extracted from images over time. This enables them to estimate properties such as direction, velocity and shape by analyzing the geometry of the STV. They solve action recognition as an object matching task, by considering the STV as a 3D object. In a related approach, Gorelick et al. [70] analyze the STV by generalizing methods for 2D shape analysis. They extract 3D shape features from a representation of the STV obtained as the solution of a Poisson equation. For classification, they use the nearest neighbor method, based on the Euclidian distance between shape features. In order to cope with variable-length movements and with the problem of temporal alignment, they define short-length superposed temporal windows within the STV and classify each of these with the method above.

A classic method used to deal with the problem of temporal alignment when performing sequence comparisons is Dynamic Time Warping (DTW) [129]. DTW is an algorithm for measuring similarity between two sequences which can vary in length or speed. It does so by searching for an optimal match between the two sequences, with some restrictions (such as monotonicity of the mapping in time). Such a match is found by performing operations of deletion / insertion, compression / expansion and substitution on the two sequences. The similarity measure between the sequences, which can be used for classification, is obtained by defining a cost of the operations performed for the matching of the two sequences. A recent paper which uses DTW for motion recognition belongs to Blackburn and Ribeiro [14]. First the human silhouette is extracted by background substraction from an image sequence. Then, the silhouettes are registered with respect to scale and translation and a distance transform is applied. Next, the sequence of silhouettes is projected to a lower-dimensional space by isometric non-linear manifold mapping (the latter being learned from training data). Finally, the trajectory in this space is classified by a nearest neighbor scheme based on the DTW matching score.

A general disadvantage of the DTW method is its ignorance of the interaction between nearby subsequences. This makes it disregard the fact that in many cases sequences that are closer in time have higher correlation than distant ones. A remedy in this respect is offered by the Hidden Markov Model (HMM) [120], which is a probabilistic temporal model. An HMM models the correlation between adjacent time instances by encapsulating a Markov process. It is a generative model, which assumes that the observed sequence has been produced by a hidden process, that performs transitions among a number of hidden states. In this context, it defines transition probabilities between pairs of hidden states and probabilities for observations given a certain state. HMMs have proved very successful for speech recognition problems and therefore many computer vision researchers decided to apply HMMs to visual recognition problems. Ahmad and Lee [3] extract the optical flow from a bounding box of the target person, together with the PCA coefficients of the human silhouette, obtained by background substraction. They use these features to classify the actions of “walking”, “running”, “raising the hand” and “bowing”, by modeling their dynamics with the aid of HMMs. Elgammal et al. [59] extract human silhouettes and perform gesture recognition via HMMs. Each gesture is represented by an HMM and the observation function of the HMMs is given by a non-parametric distribution, which enables them to associate a large number of exemplars with a small set of states. Robertson and Reid [123] build a hierarchical system for behavior understanding, where complex behavior is composed out of a set of simple actions. On the highest level, they use HMMs to model behavior. The observations of the HMMs are given by lower level features such as the trajectory, velocity and local action descriptors. The latter are obtained with the method proposed by Efros et al. [58]. Prior to feature extraction, a mean-shift tracker is used to follow the person throughout the image sequence.

A generalization of the HMM useful for behavior recognition is the Dynamic Bayesian Network (DBN) [65]. A DBN is also a probabilistic temporal model, represented as a directed graphical model of a stochastic process. It generalizes the HMM by allowing the modeling of more complex dependencies between the hidden and observed variables. In particular, it allows the arbitrary choice of the network structure at each time slice, which can be computationally expensive, but appealing in terms of flexibility. Park and Aggarwal [115] describe a method for recognizing two-person interactions using a hierarchical Bayesian network (BN). First, multiple body parts are segmented and tracked within the image sequence. Then, the poses of these parts are estimated at the low level of the BN, while the overall body pose is estimated at the high level of the BN. The interactions are classified by using a DBN which models the dynamics of body configuration changes throughout the image sequence. Luo et al. [93] propose a strategy for video analysis and recognition which uses DBNs to perform the mapping from low-level features to high-level video interpretation. The low level features are extracted from key frames detected in the video sequences, and more specifically from object silhouettes detected in these frames in a prior tracking phase.

Our framework is also formulated in terms of a DBN. In our case, the DBN permits the joining of the two processes which are considered in separation by the previous approaches: feature extraction and behavior recognition. Our proposed DBN is based on the coupling between an HMM and a probabilistic image segmentation model, used for attribute extraction from the image sequence and influenced by knowledge from the HMM.

In our approach to behavior recognition, behavior is regarded as a succession of simpler actions. As explained in the beginning of this section, this is a commonly used decomposition in behavior modeling. In this context, there are several publications which attempt to decompose behavior into simple action primitives and interpret behavior as a composition of these primitives. This topic is particularly interesting for the robotics community, in relation to the concept of “imitation learning”. In imitation learning, the aim is to develop an automatic system which can associate perceived actions to its own motor control, in order to learn, recognize and reproduce the observed actions. Therefore, research in the field is targeted at identifying a set of action primitives which enable the representation of the perceived action, as well as motor control for imitation. For instance, Billard et al. [12] use an approach based on HMMs to learn features of repetitively demonstrated movements. They use an HMM to model the motion of each joint and constrain the HMM structure so as to be able to synthesize joint trajectories of a robot. Vecchio et al. [148] use methods from the dynamical systems framework to approach the decoupling of actions into action primitives, without the constraints needed for performing action synthesis with the same representation. Such approaches could be used in conjunction with our framework, in order to identify the action primitives composing the particular behaviors involved in the targeted application.

A framework for Collaborative Segmentation and Behavior Recognition from Image Sequences

3

In this chapter, we propose a general framework for fusing bottom-up segmentation with top-down object behavior recognition over an image sequence. Such an approach is beneficial for both tasks, by enabling them to cooperate. This allows knowledge relevant to each task to aid in the resolution of the other, thus enhancing the final result of both tasks. In particular, the behavior recognition process offers dynamic probabilistic priors to guide segmentation. At the same time, segmentation supplies its results to the recognition process, ensuring that they are consistent both with prior knowledge and with new image information. The prior models are learned from training data and they adapt dynamically, based on newly analyzed images.

Our approach constitutes an important contribution to the field of behavior recognition. Namely, it offers a general solution for relieving recognition of its unconditional reliance on the uncertain results of an independent feature extraction process. Instead, the recognition takes an active part in guiding image segmentation, and invests into it all the knowledge previously acquired from training and analysis of earlier images. Furthermore, our work brings a contribution to the field of variational image segmentation. It consists of a variational formulation which incorporates multiple dynamic attribute priors, offered by the collaborating recognition process. The effectiveness of our general framework will be demonstrated in Chapter 4, via particular implementations that we have employed in the resolution of two hand gesture recognition applications.

The content of this chapter is based on material that we have published in [72, 73, 75].

3.1 Introduction

In the classical computer vision paradigm, image segmentation and object behavior recognition lie at different levels of abstraction. At a basic level, the objective of segmentation is to separate the relevant objects from the target image(s). Recognizing the behavior exhibited by such objects throughout an image sequence is a higher-level task towards comprehensive visual perception. It generally relies on prior knowledge about possible behaviors and their characteristics. Typically, the recognition problem is formulated in terms of a set of relevant attributes (e.g., color histogram, object position, orientation, shape, size, etc.), which have been extracted from the image sequence in a preceding phase (possibly, but not necessarily, by image segmentation). Thus, the phase of attribute extraction is conventionally performed separately from behavior recognition. In particular, this means that a considerable amount of information is discarded prior to the recognition phase, based on various criteria, which are not directly related to the recognition task. This happens while a wealth of knowledge regarding object behavior is left unemployed until the later stages of behavior recognition.

In this context, we pursue a joint solution to the problems of image segmentation and object behavior recognition. Clearly, a precise segmentation of the target object would greatly facilitate behavior recognition by offering access to any required object attributes. Moreover, image segmentation could be drastically improved by exploiting the knowledge which is available to the behavior recognition task. This knowledge can be used to guide the segmentation of the target object(s) in challenging conditions (e.g., images affected by noise, occlusions or cluttered background). Such a strategy can be regarded as a natural analogue of the mechanism employed by human vision, consisting in the use of previously acquired knowledge whenever there is a need to disambiguate or to complement scene information. Furthermore, from the perspective of behavior recognition, which is subject to the results provided by the attribute extraction process, an upper hand would be gained by influencing the latter towards more accurate results through the infusion of higher level knowledge.

These considerations motivate us to introduce a general framework for collaborative object segmentation and behavior recognition in image sequences. As shown in Chapter 2, the existing approaches for behavior recognition regard it as a problem of classification, using time-series of attributes extracted in a prior independent phase from the image sequence (see, e.g. [17, 57, 70, 132] and surveys like [64, 101]). Our framework is novel to the field of behavior recognition, in that it associates the two steps which are traditionally performed separately and sequentially: attribute extraction and actual behavior recognition. The purpose of this association is the collaboration of the two processes towards mutual improvement and better final results, in the spirit of the ideas presented in the previous paragraph. Furthermore, we perform attribute extraction through image segmentation, which, by delineating the object of interest, allows the flexible subsequent extraction of any attributes which are relevant for the recognition task. We formulate segmentation in a variational setting, which enables the smooth integration of both prior knowledge related to the recognition task and of specific segmentation criteria for the target images.

Variational methods offer a solid mathematical basis for the formulation and solution of many computer vision problems. In particular, as we have seen in Chapter 1, the image segmentation problem has been formulated in terms of energy minimization, allowing the seamless blending of various criteria describing the desired solution, such as smoothness, region homogeneity, edge correspondence, etc. Starting with the original active contour model [81], variational segmentation has been steadily advancing through the introduction of the Mumford-Shah model [103], the level set approach [108], geodesic active contours [28, 82, 95] and, more recently, versatile segmentation approaches such as [112, 149]. The segmentation of familiarly shaped objects in difficult cases was facilitated by the introduction of statistical shape priors into active contours [35], into level set active contours [32, 89, 126] and in the Mumford-Shah model [23, 49]. Variational methods have also been adapted to the task of object tracking (e.g., [48, 81, 111]). In this context, the coherence between consecutive frames has been exploited by variational approaches based on Kalman filtering [141], particle filtering [122], and autoregressive models [40].

Our framework fuses segmentation and behavior inference over image sequences. To our knowledge, this idea is novel in the context of variational image sequence analysis, and it capitalizes on existing developments in the use of shape priors. In previous works, segmentation has been combined with object recognition, yielding good results in the case of single, static images, both in variational [49] and non-variational [62, 84, 87, 145] settings. For tracking, [40] demonstrates the use of single-class dynamic models of motion and deformation, based on auto-regressive modeling. For image registration, [47] dynamically chooses the relevant modes of an a priori joint intensity distribution of registered image pairs, according to their proximity to the current estimated distribution. The novelty of our work is that we address the segmentation problem *over image sequences*, in a *multi-class* scenario, i.e., where the actions of the tracked object belong to classes which vary over time. Via a parallel classification strategy, we guide the segmentation dynamically towards the most likely action class at the given time. This guidance is based on learning from a training set and on accumulated evidence throughout the image sequence.

Due to its generality, our cooperative framework for the resolution of the two tasks, segmentation and behavior recognition, can be employed to resolve a wide range of applications by adapting its components and parameters according to the specific need. In particular, in Chapter 4 we illustrate the potential of our approach in two gesture recognition applications, where the cooperation of segmentation and behavior inference dramatically increases the tolerance to occlusion and background complexity present in the input image sequences.

The remainder of this chapter is organized as follows. Section 3.2 offers a general description of our framework. In Section 3.3, we formulate our joint segmentation/recognition problem in terms of a Dynamic Bayesian Network. Our strategy for joint contour estimation and behavior recognition, based on probabilistic inference through a Viterbi decoding strategy, is presented in Section 3.4. Section 3.5 details the variational formulation that we use for image segmentation. Section 3.6 presents a formal justification of an approximation used to derive our Viterbi decoding strategy, and of our variational competition approach. In Section 3.7, we explain the advantage of using our approach for competition among

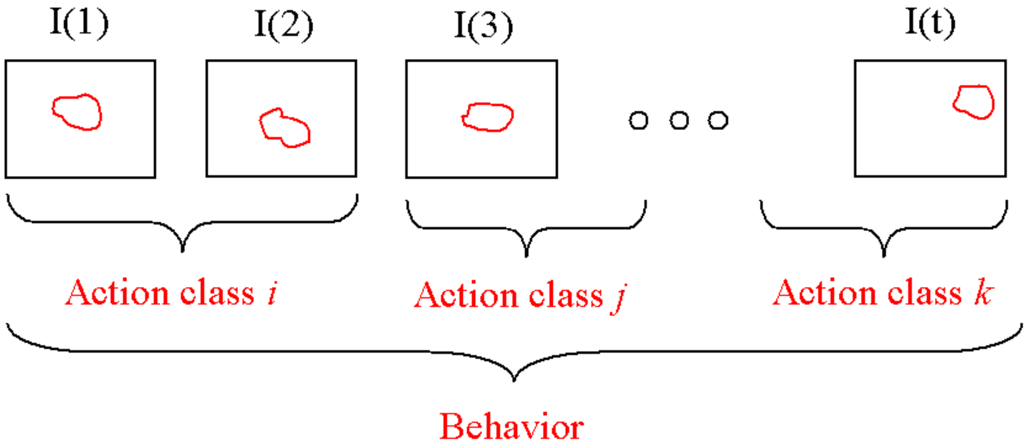


Figure 3.1 — Recognition of object behavior in image sequences. The recognized behavior is expressed as the sequence of action classes corresponding to each time instant.

multiple priors, compared to a well-known approach proposed by Cremers et al. [46, 49]. Section 3.8 explains how the parameters of our model can be learned from adequate training sequences. In Section 3.9, we introduce an extension of our model which allows the specific treatment of behaviors belonging to a finite set of behavior types. Section 3.10 summarizes our approach in an algorithmic setting and Section 3.11 concludes the chapter.

3.2 General Description of the Framework

Limiting the scope of our work to single object segmentation and behavior recognition, we can define "behavior" as the temporal evolution of the object, observed in the image sequence. Now, let us consider object behavior throughout an arbitrarily long image sequence as being composed of a set of basic primitives, that we call actions. Then, the recognition of object behavior translates to assigning each object evolution instance the appropriate action class, as illustrated in Fig. 3.1. The recognized behavior is given by the succession of these action classes along the image sequence. At the basis of recognition lies prior knowledge about the possible action classes, their characteristics and the typical ways in which they associate to compose behaviors.

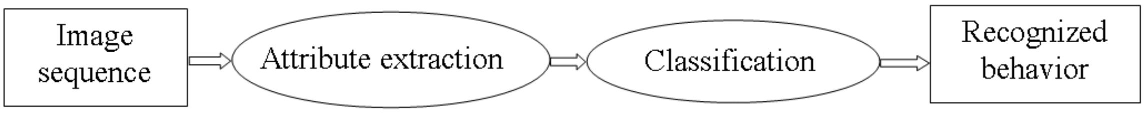


Figure 3.2 — The typical approach to behavior recognition, composed of two sequential steps: attribute extraction and classification, the latter yielding the recognized behavior.

The approaches for behavior recognition encountered in the literature typically consist of two *separate* steps, performed *sequentially* (see Fig. 3.2): the extraction of attributes from the target image sequence and the classification of the resulting attribute time-series, leading to the recognition of the exhibited behavior. In contrast to these approaches, we propose to fuse the two steps, by performing them *simultaneously* and *in cooperation*, as shown in Fig. 3.3.

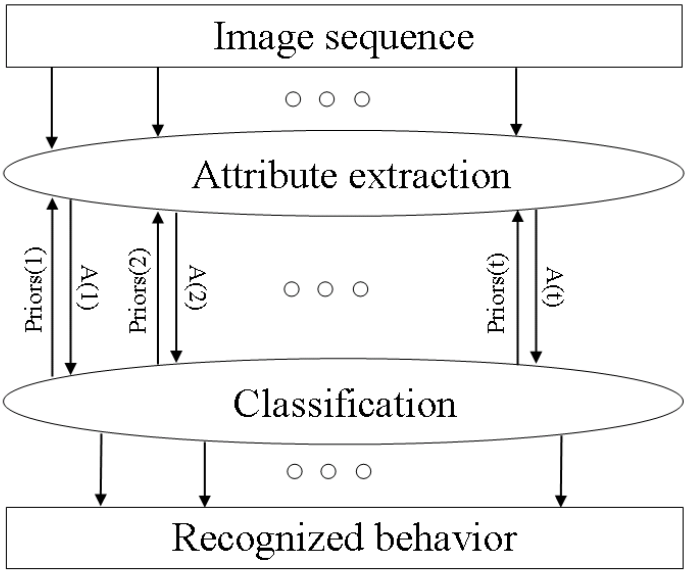


Figure 3.3 — Our framework for behavior recognition, based on the cooperation of the two processes: attribute extraction and classification. Classification offers probabilistic attribute priors to guide attribute extraction. In turn, the attribute extraction process supplies the newly detected attribute values to classification, ensuring they are consistent with prior knowledge.

The attribute extraction process is performed through variational image segmentation, which is guided towards the most likely target object by attribute priors supplied by the classification. Our classification strategy is based on probabilistic inference. This means that we use a learned model to answer the question of frame classification into one of several possible action classes. To this end, we use dynamic models of behavior, which adapt to incorporate information (attributes) from new images analyzed by segmentation. These dynamic probability models encapsulate typical behaviors and are learned from training data during an initial training procedure, performed before the application of our cooperative framework to new image sequences targeted for behavior recognition. After the training phase, segmentation and probabilistic inference are run cooperatively in an interleaved manner throughout a new test image sequence. More precisely, for each image, an inference step is performed, generating probabilistic prior attribute models corresponding to each of the possible action classes. These are used by the ensuing segmentation to identify the most likely objects in the current image and subsequently provide their attributes to the next inference step. The procedure continues in the same fashion up to the end of the image sequence. At any time instance along the sequence, the most likely succession of action classes up to that instance can be retrieved from the inference process. This makes our framework suitable for the online processing of continuous behavior sequences of arbitrary length.

Before proceeding with the detailed illustration of the two halves of our framework — classification by behavior inference and segmentation — let us define the generic term "attribute", relative to our framework where attributes are extracted by segmentation. In this context, the "attribute" designates a vector which encapsulates visual properties of an object, definable as a functional $f_A(I, C)$ of the image I and of the object's segmenting contour C (f_A is assumed to be differentiable with respect to C). This definition includes many object properties computable with boundary- and region-based functionals, such as position, orientation, average intensity/color or higher order statistics describing texture. Such flexibility in the choice of the extracted attributes makes our framework adaptable to the needs of a wide range of behavior recognition applications.

3.3 A Model for the Joint Segmentation and Behavior Recognition of Image Sequences

3.3.1 Motivation

The aim of our framework is to jointly segment the object of interest from an image sequence and to recognize the corresponding behavior. We model this joint segmentation/recognition problem using a Dynamic Bayesian Network (DBN) based on a Hidden Markov Model (HMM). A DBN is a probabilistic temporal model that represents a sequence of variables. An HMM is a particular type of DBN, which associates a sequence of discrete states to a sequence of observations — in our case, a sequence of images. Each state is characterized by a probability distribution, often called the emission distribution, which gives the probability of an observation while being in the respective state. The evolution of the states with time is controlled by a transition distribution, which represents the probability of switching to a certain state given the current state. The states are considered as hidden and the only evidence about them is given by the sequence of observations. An approach based on an HMM is particularly appealing in the context of behavior recognition because a discrete state is a natural representation of a behavior component, that we denote as action. The transition distribution then models the fact that, inside a particular behavior, certain sequences of actions are more likely to be observed than others.

3.3.2 The Model

Given an image sequence $I_{1:T} = \{I_1, I_2, \ldots, I_T\}$, our segmentation task translates to finding the target object's contour C_t in each image I_t, yielding the contour sequence $C_{1:T} = \{C_1, C_2, \ldots, C_T\}$. Similarly, behavior recognition amounts to determining the action class s_t which corresponds to the observed image I_t, yielding the action class sequence $s_{1:T} = \{s_1, s_2, \ldots, s_T\}$. The action classes that compose the behaviors under study belong to a finite set $S = \{S_1, S_2, \ldots, S_M\}$. The different behaviors (and their component actions) are distinguished in terms of the object attributes A_t, which are extracted from the images I_t

by means of segmentation. Formally, this can be written as $A_t = f_A(I_t, C_t)$, where f_A is a function which associates to a given image I_t and contour C_t the corresponding extracted attribute A_t.

In this context, we model the joint segmentation and behavior recognition problem using the Dynamic Bayesian Network shown in Fig. 3.4. In this figure, we have represented the model corresponding to two time slices — $t-1$ and t — the dots implying that the DBN structure and parameters repeat in a similar fashion, starting from the first time slice, up to the one corresponding to the last image in the sequence that it models. Our model is based on coupling an HMM — whose hidden state at time t is given by the action class s_t — with a probabilistic generative segmentation model, where the image I_t depends on the contour C_t and the attribute A_t. The coupling of the two models at each time t is realized through the attribute A_t. We represent observed variables by shaded nodes (the images I_t, $t = 1..T$) and hidden variables by clear nodes (the classes s_t, the attributes A_t and the contours C_t, $t = 1..T$). Moreover, we depict discrete variables by square nodes (the classes s_t, $t = 1..T$) and continuous variables by circular ones (the attributes A_t, the contours C_t and the images I_t, $t = 1..T$).

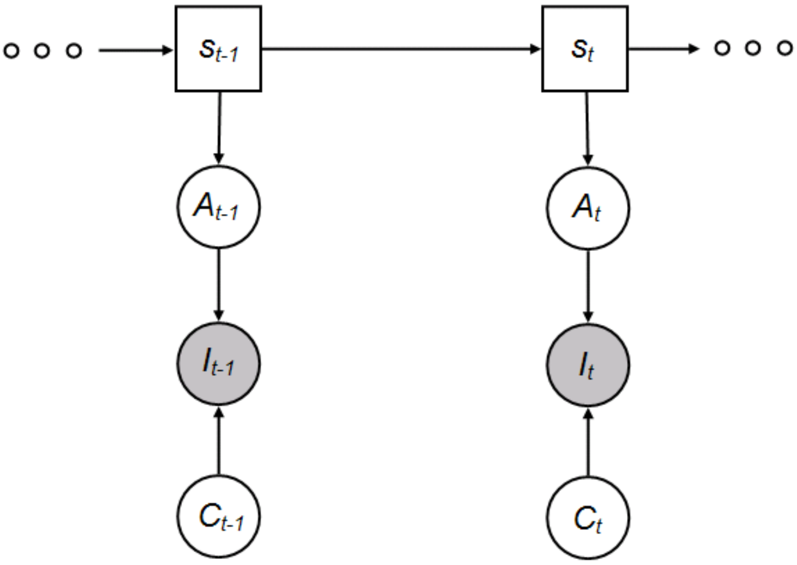


Figure 3.4 — The Dynamic Bayesian Network supporting our joint segmentation / behavior recognition framework. This model can be regarded as containing an HMM (in the upper half), coupled with a probabilistic segmentation model (in the lower half). For time slice t, the hidden state of the HMM is given by action class s_t. Within the generative segmentation model, the image I_t is dependent on the contour C_t and the attribute A_t. The observation at time t is given by the image I_t. We depict hidden variables by clear nodes and observed variables by shaded nodes. The square nodes designate discrete variables, whereas circular ones designate continuous variables.

According to the DBN represented in Fig. 3.4, our model is characterized by the following joint variable distribution:

$$P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}) = \prod_{t=1}^{T} P(I_t|A_t, C_t)P(C_t)P(A_t|s_t)P(s_t|s_{t-1}), \tag{3.1}$$

where $P(s_1|s_0) \equiv P(s_1)$ is the initial action class distribution. In the following, we explain the assumptions underlying our model and we detail each of the probability factors from the right-hand side product in (3.1).

Our model relies on the first order Markov assumption, namely that the action class at time t only depends on the action class at time $t-1$, being independent with respect to the action classes previous to time $t-1$:

$$P(s_t|s_{t-1}, s_{t-2}, \ldots) \equiv P(s_t|s_{t-1}). \tag{3.2}$$

The right-hand side of (3.2), which is part of our model (3.1), is considered to be independent of time, and thus definable in terms of the set of action class transition probabilities $T = \{t_{ij}\}$:

$$P(s_t = S_j|s_{t-1} = S_i) = t_{ij}, \quad i, j = 1..M, \tag{3.3}$$

under the standard stochastic constraints:

$$\begin{gathered} t_{ij} \geq 0 \\ \sum_{j=1}^{M} t_{ij} = 1. \end{gathered} \tag{3.4}$$

The initial action class distribution, corresponding to the action class of the first image in a sequence, is given by $\pi = \{\pi_i\}$, with

$$\pi_i = P(s_1 = S_i), \quad i = 1..M. \tag{3.5}$$

In order to incorporate the attributes A_t, the object contour C_t and the image I_t in our probabilistic model, we need to treat those quantities as random variables. Within our DBN, which is founded on an HMM, this is achieved by defining the joint distribution of these variables given the action class s_t, that is, $P(I_t, C_t, A_t|s_t)$. Directly working with such a joint distribution is in general too complicated. The model can often be made more tractable by considering a simpler factorized distribution, where some of the dependencies between the variables are removed. In our model, we propose to use a joint distribution of the form

$$P(I_t, C_t, A_t|s_t) = P(I_t|A_t, C_t)\, P(C_t)\, P(A_t|s_t). \tag{3.6}$$

In our framework, the attributes A_t represent the essential characteristics of the object captured in image I_t, which are relevant for the recognition task. The prior knowledge we have about these attributes, associated to a particular action class, is given by $P(A_t|s_t)$, which represents the probability of the attributes A_t given the action class s_t. Of course, the most suitable model for this probability depends on the application to be solved and on the type of attributes that were chosen. Thus, we let the modeling of this probability constitute one of the "degrees of freedom" of our framework, to be performed according to the application at hand. For notation simplification, we denote the attribute probability given an action class S_i by:

$$P_i(A_t) = P(A_t|s_t = S_i). \tag{3.7}$$

To support cooperation with the segmentation process, we only require that these probabilities be modeled by functions $P_i(A_t)$ which are differentiable with respect to A_t. Examples of models for this probability will be offered in Chapter 4, where we present implementations of our framework for particular applications.

The probabilities $P(C_t)$ and $P(I_t|A_t, C_t)$ in (3.1) constitute a probabilistic segmentation model, that we will translate into a variational segmentation formulation. In this context, we would like to note that the object contour C is a continuous function, belonging to an infinite-dimensional space. Generally, the modeling of probability distributions on infinite-dimensional spaces is an open issue. Thus, in practice, we consider a finite-dimensional representation of the contour, obtained by sampling over a regular grid.

In our framework, the prior probability of the contour $P(C_t)$ is another free parameter, which gives us the possibility to include (application-dependent) a priori knowledge about the target object contour, which is independent of the action class. As we have seen in Chapter 2, a common choice for this probability in the variational segmentation community favors a short length $|C_t|$ of the segmenting contour:

$$P(C_t) \propto e^{-\nu|C_t|}, \quad \nu > 0. \tag{3.8}$$

Moreover, $P(I_t|A_t, C_t)$ corresponds to a generative image formation model. This model states that, given a set of prior attributes A_t and a prior contour C_t, an image I_t can be obtained by sampling from the distribution $P(I_t|A_t, C_t)$. In other words, this means that we focus on the attributes and object contour only, and consider all the other properties of the image as resulting from random variations. The distribution $P(I_t|A_t, C_t)$ represents the probability of observing image I_t, given that C_t is the boundary of the object of interest and $A_t = f_A(I_t, C_t)$ are the attributes extracted from the image via the function f_A. Since f_A is a deterministic function of I_t and C_t, we need to give it a probabilistic interpretation in order to be able to incorporate it into our model. A simple approach is to consider that the probability of observing an image I_t whose extracted contour is C_t and whose extracted attributes A_t are different from $f_A(I_t, C_t)$, is zero. Formally, this can be achieved by defining

$$P(I_t|A_t, C_t) \propto \delta(A_t - f_A(I_t, C_t))\, e^{-E_{\text{image}}(I_t, C_t)}, \tag{3.9}$$

where δ represents a Dirac distribution, which selects the images with the right attributes. Moreover, E_{image} is a free parameter of our framework, given by a variational segmentation energy, which expresses image-based constraints on the contour. It can be made up of any boundary- or region-based energy terms suitable for the application at hand (such as the ones adopted in [30] or [112], presented in Chapter 2). Denoting by $\Omega \subset \mathbb{R}^2$ the image domain and by $\omega \subset \Omega$ — the region inside C_t, a typical example for E_{image} is given by assuming the values of the image feature values $I(x, y)$ (which can be scalar or vectorial) to be independent and identically distributed samples of two independent random processes, corresponding to the object and background region, respectively:

$$\begin{aligned} E_{\text{image}}(I_t, C_t) = &\iint_{\omega} - \log P(I_t(x, y)|(x, y) \in \omega)\, dx\, dy \\ &+ \iint_{\Omega \setminus \omega} - \log P(I_t(x, y)|(x, y) \in \Omega \setminus \omega)\, dx\, dy. \end{aligned} \tag{3.10}$$

A common modeling choice for the region probabilities $P(I_t(x, y)|(x, y) \in \omega)$ and $P(I_t(x, y)|(x, y) \in \Omega \setminus \omega)$ is the Gaussian distribution. Concrete modeling examples for the application-dependent parameters of our framework, i.e., $P(A_t|s_t)$, $P(C_t)$ and $E_{\text{image}}(I_t, C_t)$, will be offered in Chapter 4.

3.4 Joint Segmentation and Behavior Recognition in Image Sequences

Our joint segmentation / behavior recognition problem can be formulated in probabilistic terms as the task of finding the contours $C_{1:T}$ and the action classes $s_{1:T}$ whose probability given the observed images $I_{1:T}$ is maximum:

$$(s^*_{1:T}, C^*_{1:T}) = \arg\max_{\substack{s_{1:T} \\ C_{1:T}}} P(C_{1:T}, s_{1:T}|I_{1:T}). \tag{3.11}$$

This can be equivalently written as:

$$(s^*_{1:T}, C^*_{1:T}) = \arg\max_{\substack{s_{1:T} \\ C_{1:T}}} P(C_{1:T}, s_{1:T}, I_{1:T}), \tag{3.12}$$

where $P(I_{1:T}, C_{1:T}, s_{1:T})$ is obtained by integrating the joint distribution given by Eq. 3.1 over the attributes $A_{1:T}$, i.e.,

$$P(I_{1:T}, C_{1:T}, s_{1:T}) = \int_{A_{1:T}} P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}). \tag{3.13}$$

Some insight on how to solve Eq. 3.12 can be gained by first considering the problem of finding the likelihood of the most likely configuration $(s^*_{1:T}, C^*_{1:T})$:

$$P(I_{1:T}, C^*_{1:T}, s^*_{1:T}) = \max_{\substack{s_{1:T} \\ C_{1:T}}} P(I_{1:T}, C_{1:T}, s_{1:T}). \tag{3.14}$$

The structure of the DBN of Fig. 3.4 suggests that, considering a time moment $t \in$

$\{1, \dots, T-1\}$, the right-hand side of Eq. 3.20 can be written as:

$$\begin{aligned}
&\max_{\substack{s_{1:T}\\C_{1:T}}} P(I_{1:T}, C_{1:T}, s_{1:T})\\
&= \max_{\substack{s_{1:T}\\C_{1:T}}} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T} | I_{1:t}, C_{1:t}, s_{1:t+1})\, P(I_{1:t}, C_{1:t}, s_{1:t+1})\\
&= \max_{\substack{s_{1:T}\\C_{1:T}}} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T} | s_{t+1})\, P(I_{1:t}, C_{1:t}, s_{1:t+1})\\
&= \max_{\substack{s_{t+1:T}\\C_{t+1:T}}} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T} | s_{t+1}) \max_{\substack{s_{1:t}\\C_{1:t}}} P(s_{t+1} | I_{1:t}, C_{1:t}, s_{1:t})\, P(I_{1:t}, C_{1:t}, s_{1:t})\\
&= \max_{\substack{s_{t+1:T}\\C_{t+1:T}}} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T} | s_{t+1}) \max_{s_t} P(s_{t+1} | s_t) \max_{\substack{s_{1:t-1}\\C_{1:t}}} P(I_{1:t}, C_{1:t}, s_{1:t}). \qquad (3.15)
\end{aligned}$$

For the second equality, we used the fact that, according to the DBN of Fig. 3.4, the future observations $I_{t+1:T}$, contours $C_{t+1:T}$ and actions classes $s_{t+2:T}$ are independent of any past quantity once s_{t+1} is known. Similarly, for the fourth equality, we used the fact that s_{t+1} is independent of the past images, contours and action classes once s_t is known. The probability $P(I_{1:t}, C_{1:t}, s_{1:t})$ from Eq. 3.15 can be written as:

$$\begin{aligned}
P(I_{1:t}, C_{1:t}, s_{1:t}) &= P(I_t, C_t | I_{1:t-1}, C_{1:t-1}, s_{1:t})\, P(I_{1:t-1}, C_{1:t-1}, s_{1:t})\\
&= P(I_t, C_t | s_t)\, P(s_t | I_{1:t-1}, C_{1:t-1}, s_{1:t-1})\, P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1})\\
&= P(I_t, C_t | s_t)\, P(s_t | s_{t-1})\, P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}) \qquad (3.16)
\end{aligned}$$

The second equality is motivated by the fact that the image I_t and the contour C_t are independent of any past quantities once s_t is given. Likewise, the third equality results from the fact that the action class s_t is independent of any past quantities once s_{t-1} is given. Using the result of Eq. 3.16, we can express the maximization over $s_{1:t-1}$ and $C_{1:t}$ in (3.15) as:

$$\max_{\substack{s_{1:t-1}\\C_{1:t}}} P(I_{1:t}, C_{1:t}, s_{1:t}) = \max_{C_t} P(I_t, C_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}) \max_{\substack{s_{1:t-2}\\C_{1:t-1}}} P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}). \qquad (3.17)$$

This formulation prompts us to the definition of the quantity $\delta_t(s_t)$ as:

$$\delta_t(s_t) = \max_{\substack{s_{1:t-1}\\C_{1:t}}} P(I_{1:t}, C_{1:t}, s_{1:t}). \qquad (3.18)$$

According to Eq. 3.17, $\delta_t(s_t)$ can be computed with the recursive formula:

$$\delta_t(s_t) = \max_{C_t} P(I_t, C_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1})\, \delta_{t-1}(s_{t-1}), \qquad (3.19)$$

which is initialized by setting $\delta_0(s_0) = 1$. Therefore, we can obtain the likelihood of the most likely configuration $(s^*_{1:T}, C^*_{1:T})$, defined by (3.20), by recursively estimating $\delta_t(s_t)$ for each time step $t \in \{1, \dots, T\}$ and each action class $s_t \in S$, and then maximizing $\delta_T(s_T)$ over the action class s_T:

$$P(I_{1:T}, C^*_{1:T}, s^*_{1:T}) = \max_{s_T} \delta_T(s_T). \qquad (3.20)$$

The recursive formulation used to calculate $\delta_t(s_t)$ allows us to exhaustively explore all the possible action class sequences $s_{1:T}$ in order to find the one which, together with its associated estimated contours $C^*_{1:T}$, achieves the global maximization of $P(I_{1:T}, C_{1:T}, s_{1:T})$. This optimal action class sequence, defined by Eq. 3.12, can be retrieved by storing, at each time step t and for each action class s_t, the action class s_{t-1} which maximizes the right-hand side of Eq. 3.19. Formally, if we denote by $\psi_t(s_t)$ this latter quantity, then we have

$$\psi_t(s_t) = \arg\max_{s_{t-1}} P(s_t|s_{t-1})\,\delta_{t-1}(s_{t-1}). \tag{3.21}$$

Therefore, the most likely action class sequence $s^*_{1:T}$ can be found by applying iteratively, and backward in time, the formulae:

$$\begin{aligned} s^*_T &= \arg\max_{s_T} \delta_T(s_T), \\ s^*_t &= \psi_{t+1}(s^*_{t+1}), \quad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{3.22}$$

Equations 3.19 and 3.22 form a Viterbi decoding algorithm [150] adapted to our model. A difference between our formulation and the one generally encountered in the HMM literature [120], is the presence of the additional maximization over the hidden variable C_t in Eq. 3.19.

According to Eq. 3.19, once $s^*_{1:T}$ has been obtained, the most likely contour sequence $C^*_{1:T}$, defined by Eq. 3.12, is given by

$$C^*_t = \arg\max_{C_t} P(I_t, C_t|s^*_t). \tag{3.23}$$

Using Eq. 3.9, $P(I_t, C_t|s_t)$ can be written as:

$$\begin{aligned} P(I_t, C_t|s_t) &= \int_{A_t} P(I_t, A_t, C_t|s_t) \\ &= \int_{A_t} P(I_t|A_t, C_t)\,P(C_t)\,P(A_t|s_t) \\ &\propto \int_{A_t} \delta(A_t - f_A(I_t, C_t))\,e^{-E_{\text{image}}(I_t,C_t)}\,P(C_t)\,P(A_t|s_t) \\ &\propto e^{-E_{\text{image}}(I_t,C_t)}\,P(C_t)\,P(A_t = f_A(I_t, C_t)|s_t). \end{aligned} \tag{3.24}$$

Using a Dirac distribution centered on the attributes A_t in $P(I_t|A_t, C_t)$ proves to be particularly handy here because it allows us to easily integrate over A_t.

The maximization over C_t in Eq. 3.19 requires the computation of the *locally* most likely contour $C^*(s_t)$ for each action class s_t:

$$C^*(s_t) = \arg\max_{C_t} P(I_t, C_t|s_t). \tag{3.25}$$

However, since the estimation of $C^*(s_t)$ needs to be performed by image segmentation, the time costs of repeating the segmentation procedure for each action class s_t can be prohibitive. We therefore prefer to choose an alternative solution, where the segmentation of the image I_t is performed only once. Such a solution is more desirable if we wish our framework to scale well with an increasing number of action classes. A possible approach is to approximate $\delta_t(s_t)$, given by Eq. 3.19, by

$$\tilde{\delta}_t(s_t) = P(I_t, \tilde{C}_t^* | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}) \, \tilde{\delta}_{t-1}(s_{t-1}), \tag{3.26}$$

or equivalently

$$\tilde{\delta}_t(s_t) = P(I_t, \tilde{C}_t^* | s_t) \, w_t(s_t), \tag{3.27}$$

where we define

$$w_t(s_t) = \max_{s_{t-1}} P(s_t | s_{t-1}) \, \tilde{\delta}_{t-1}(s_{t-1}). \tag{3.28}$$

In Eq. 3.26 and 3.27, $\tilde{C}_t^*$ is an approximation of the most likely contour C_t^* (Eq. 3.23), obtained from a single segmentation of the image I_t, and is given by

$$\tilde{C}_t^* = \arg\max_{C_t} \max_{s_t} P(I_t, C_t | s_t) \, w_t(s_t), \tag{3.29}$$

where $w_t(s_t)$ is defined in Eq. 3.28. Equation 3.29 shows that we make an approximation of the true most likely contour C_t^* for image I_t, based on the currently most likely action class s_t, in the light of past evidence accumulated in the δ quantities and of the new image information given by I_t. This constitutes a "greedy" technique, making a final and (most-likely) locally optimum solution based on the current existing information. The details of our segmentation method implementing (3.29) are presented in the next section. A formal justification of our approximation of the true most likely contour C_t^* from(3.23) by the locally most likely contour $\tilde{C}_t^*$ in (3.29) is given in Section 3.6.

The first time step of our recursive formulation (3.26) reads

$$\tilde{\delta}_1(s_1) = P(I_1, \tilde{C}_1^* | s_1) \, P(s_1). \tag{3.30}$$

Here $\tilde{C}_1^*$ is obtained by the segmentation of the first image I_1 of the sequence $I_{1:T}$, for which no classification information regarding the current sequence is available yet ($w_1(s_1) = P(s_1)$).

Given the fact that our segmentation method is quite sensitive to its initial conditions (as is the case with all variational segmentation methods) and also the fact that we use the final segmentation contour of one image as the initial contour for the next image, it is desirable to obtain a good segmentation of the first image in the sequence. Therefore, we leave the particular segmentation method employed for the first image of a sequence as a free parameter of our framework, to be chosen depending on the application. Along the lines of our original formulation, one option is to perform this segmentation automatically, using (3.29), with $w_1(s_1) = P(s_1)$, and the variational segmentation scheme that we propose in the following section. Alternatively, one can perform the segmentation once for each possible value of s_1, as in (3.25) and then choose the most likely contour for the first image as the one corresponding to the value of s_1 which maximizes $\delta_1(s_1)$ given by (3.19). The segmentation in this case can also be performed by a simplification of our variational scheme presented in the next section. The most reliable method, but also the most time-consuming for the human operator, is the manual segmentation of the first image. Irrespective of the particular method that is chosen, we consider for the moment that a satisfactory segmentation C_1 of I_1 is available. Concrete segmentation models of the initial image are provided in Chapter 4.

Similarly to our initial formulation of the Viterbi decoding algorithm, in order to be able to retrieve the optimal action class sequence $s^*_{1:T}$ by backtracking, we need to store the argument fulfilling the maximization from the computation of $\tilde{\delta}_t(s_t)$ (3.26), for each time slice $t > 1$ and each value of s_t, using:

$$\psi_t(s_t) = \arg\max_{s_{t-1}} P(s_t|s_{t-1})\, \tilde{\delta}_{t-1}(s_{t-1}), \quad s_t \in S. \tag{3.31}$$

Then, the optimal action class sequence $s^*_{1:T}$ can be obtained by backtracking from $\psi_t(s_t)$, using the equations

$$\begin{aligned} s^*_T &= \arg\max_{s_T} \tilde{\delta}_t(s_T), \\ s^*_t &= \psi_{t+1}(s^*_{t+1}), \qquad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{3.32}$$

3.5 Variational Segmentation Formulation for Contour Estimation

Statistical interpretations of variational segmentation methods were offered, among others, in [25, 39, 40, 50, 112, 166]. In the same spirit, we translate our probabilistic formulation for attribute and contour estimation (3.29) into a variational segmentation formulation. Combining Eq. 3.29 and 3.24, we obtain:

$$\begin{aligned} \tilde{C}^*_t &= \arg\max_{C_t} \left(\max_{s_t} e^{-E_{\text{image}}(I_t,C_t)}\, P(C_t)\, P(A_t = f_A(I_t, C_t)|s_t)\, w_t(s_t) \right) \\ &= \arg\max_{C_t} \left(e^{-E_{\text{image}}(I_t,C_t)}\, P(C_t)\, \max_{s_t} P(A_t = f_A(I_t, C_t)|s_t)\, w_t(s_t) \right) \end{aligned} \tag{3.33}$$

Towards a variational segmentation formulation, we equate the maximization with respect to the contour C_t in (3.33) with the minimization with respect to C_t of the negative logarithm of the right-hand side quantity:

$$\tilde{C}^*_t = \arg\min_{C_t} \left(E_{\text{image}}(I_t, C_t) \;- \log P(C_t) - \min_{s_t} \log \left(P(A_t = f_A(I_t, C_t)|s_t)\, w_t(s_t) \right) \right). \tag{3.34}$$

By identifying the first term of the right-hand side with an image-dependent energy term, the second one with a contour-dependent energy term, and the third one with an energy term embodying prior information offered by the recognition process, we can formulate our total segmentation energy as the sum of three energies:

$$E(C_t, \mathcal{L}, I_t) = E_{\text{image}}(I_t, C_t) + \nu E_{\text{contour}}(C_t) + \alpha E_{\text{prior}}(C_t, \mathcal{L}, I_t). \tag{3.35}$$

Here ν and α are positive constants which balance the contributions to segmentation of the three terms and $\mathcal{L} = (L_1, \dots L_M)$ is a set of labels, which serves in the implementation of the minimization with respect to the class s_t from (3.34), as will be shown in the following.

As explained in Section 3.3, the image-dependent energy term $E_{\text{image}}(I_t, C_t)$ can contain any region- or boundary-based energy term which suits the application to be solved. A generic example of such energy, modeling the object and background pixels by two different probability distributions, is given by (3.10). Moreover, the contour dependent term $E_{\text{contour}}(C_t)$ expresses a priori knowledge regarding the contour, generally including smoothness constraints on the contour. An example is the term limiting contour length, obtained by choosing $P(C_t)$ as in (3.8), that is:

$$E_{\text{contour}}(C_t) = |C_t|. \tag{3.36}$$

The third term of the right-hand side of (3.34) is the one which incorporates prior information, provided by the recognition process. We include the minimization implied by this term within the variational segmentation formulation by means of a competition approach, motivated by [46, 49]. To this end, we consider the following prior energy term:

$$E_{\text{prior}}(C_t, \mathcal{L}, I_t) = -\sum_{i=1}^{M} \log\left(P(A_t|s_t = S_i)\, w_t(S_i)\right)\, L_i^2 + \beta\left(1 - \sum_{i=1}^{M} L_i^2\right)^2, \tag{3.37}$$

where $A_t = f_A(I_t, C_t)$. The set of labels $\mathcal{L} = (L_1, \dots, L_M)$ controls the contribution to segmentation of the attribute prior information corresponding to each action class S_i, according to its respective probability $P(A_t|s_t = S_i)\, w_t(S_i)$. The label L_i is a scalar variable that varies continuously between 0 and 1 during energy minimization, according to the corresponding gradient descent evolution equation. The evolution of a label converges either to 1 (for the winning prior class S_i, corresponding to the probability $P(A_t|s_t = S_i)\, w_t(S_i)$ that has been maximized through segmentation, since it has been present in the energy (3.37)) or to 0 (for the other priors, whose contribution has thus been annulled). Competition among priors is enforced by the constraint that the label factors should sum to 1, introduced by the term $\beta\,(1-\sum_{i=1}^{M} L_i^2)^2$ in energy (3.37). Here β is a Lagrange multiplier, updated at each energy minimization step to ensure that $(1 - \sum_{i=1}^{M} L_i^2)^2 \approx 0$, as will be explained in the following. A similar technique has been applied to solve the problems of vacuum and overlap for multi-phase image segmentation in [165]. The competition between the attribute priors of the different action classes during energy minimization means that the final estimated segmenting contour C_t will be obtained by the influence of the most likely action class, in light of image evidence. Therefore, the minimization of our proposed total energy (3.35) with respect to the labels $\mathcal{L}$, can be considered as the equivalent of the maximization with respect to the class s_t from (3.90). Naturally, since this minimization is performed through gradient descent, only a local minimum of the energy with respect to $\mathcal{L}$ will be attained.

To better understand the nature of the prior information that we infuse into our segmentation model (3.35), we analyze the factors of the probability product $P(A_t|s_t = S_i)\, w_t(S_i)$ corresponding to each action class S_i:

- $P(A_t|s_t = S_i)$ is the attribute probability function offered by the action class S_i as an indication of the detected object's attributes in image I_t, and

- $w_t(S_i)$ is the relative prior confidence that the image at time t will correspond to class S_i, given by the recognition scheme on the account of past evidence $I_{1..t-1}$, accumulated into $\tilde{\delta}_{t-1}(s_{t-1})$:

  $$w_t(S_i) = \max_{s_{t-1}} P(s_t = S_i|s_{t-1})\,\tilde{\delta}_{t-1}(s_{t-1}). \tag{3.38}$$

  Given our introduced approximation for $\delta_t(s_t)$ from (3.19) as $\tilde{\delta}_t(s_t)$ (3.26), this quantity approximates the maximum probability that the image I_t belongs to action class S_i, after having observed images $I_{1..t-1}$.

Therefore, the product of the two factors above encapsulates the maximum amount of a priori knowledge regarding the action class and attribute at moment t, available within the recognition scheme before the segmentation of image I_t.

We minimize the total energy (3.35) simultaneously with respect to the segmenting contour C_t and the labels $\mathcal{L}$ using the calculus of variations and gradient descent. The contour C_t is driven by image forces (region homogeneity, gradients, etc.) due to $E_{\text{image}}(I_t, C_t)$, smoothing forces due to $E_{\text{contour}}(C_t)$ and by the competing attribute priors of each action class, due to $E_{\text{prior}}(C_t, \mathcal{L}, I_t)$:

$$\frac{\partial C_t}{\partial \tau} = -\frac{\partial E_{\text{image}}(I_t, C_t)}{\partial C_t} - \nu \frac{\partial E_{\text{contour}}(C_t)}{\partial C_t} - \alpha \frac{\partial E_{\text{prior}}(C_t, \mathcal{L}, I_t)}{\partial C_t}. \tag{3.39}$$

Here τ is the artificial time of variable evolution. The first variations of the energies $\frac{\partial E_{\text{image}}(I_t,C_t)}{\partial C_t}$ and $\frac{\partial E_{\text{contour}}(C_t)}{\partial C_t}$ can be derived through the calculus of variations for the particular chosen forms of $E_{\text{image}}(I_t, C_t)$ and $E_{\text{contour}}(C_t)$, respectively. The third term of (3.39) can be written as:

$$\frac{\partial E_{\text{prior}}(C_t, \mathcal{L}, I_t)}{\partial C_t} = -\sum_{i=1}^{M} \frac{L_i^2}{P(A_t|s_t = S_i)} \; \frac{\partial P(A_t|s_t = S_i)}{\partial A_t} \, \frac{\partial f_A(I_t, C_t)}{\partial C_t}, \tag{3.40}$$

where $A_t = f_A(I_t, C_t)$ and the derivatives $\partial P(A_t|s_t = S_i)/\partial A_t$ and $\partial f_A(I_t, C_t)/\partial C_t$ are computed according to the particular probability model and attribute employed.

Through gradient descent derivation, we obtain the following evolution equations for the labels L_i:

$$\frac{\partial L_i}{\partial \tau} = L_i \left(\log\big(P(A_t|s_t = S_i)\, w_t(S_i)\big) + 2\beta \left(1 - \sum_{i=1}^{M} L_i^2\right)\right), \qquad i = 1..M. \tag{3.41}$$

The labels are initialized with equal values, so that $(1 - \sum_{i=1}^{M} L_i^2)^2 \approx 0$, for instance by

$$L_i = 1/\sqrt{M} - \epsilon_L, \quad \epsilon_L = 10^{-5}. \tag{3.42}$$

The update equation for the Lagrange multiplier β is deduced by imposing constancy of the constraint over time: $d(1-\sum_{i=1}^{M} L_i^2)^2/d\tau = 0$. This yields the following update equation:

$$\beta = \frac{\sum_{i=1}^{M} L_i^2 \, \log\big(P(A_t|s_t = S_i)\, w_t(S_i)\big)}{2\sum_{i=1}^{M} L_i^2 \big(\sum_{i=1}^{M} L_i^2 - 1\big)}. \tag{3.43}$$

Thus, the segmentation of an image I_t, $t > 0$ comprises the following steps:

1. Initialize contour C_t with the final estimated contour of the previous image: $C_t = \tilde{C}_{t-1}^*$.
2. Initialize labels L_i using (3.42).
3. **while** (not converged(C_t))
    - (a) Perform one contour evolution step given by (3.39).
    - (b) Update the Lagrange multiplier β, using (3.43).
    - (c) Perform one evolution step for each label L_i, $i = 1..M$ using (3.41).
4. **end**
5. $\tilde{C}_t^* = C_t$.

Like all variational segmentation methods, the practical implementation of our proposed segmentation formulation implies the use of appropriate numerical schemes for the discretization of the evolution equations (3.39) and (3.41). Examples of such schemes will be offered for the concrete implementations of our framework in Chapter 4. The convergence with respect to the contour C_t, mentioned in the segmentation strategy above, can be tested by verifying whether the contour rate of change falls below a predefined threshold (meaning that it remains approximately constant).

3.6 Formal Derivation of our Competition-Between-Priors Scheme

The competition-between-priors strategy that we used in Eq. 3.37 to solve Eq. 3.33 resulted from our desire to approximate the true most likely contour C_t^*, given by Eq. 3.23, by the *locally* most likely contour $\tilde{C}_t^*$, given by Eq. 3.29. In this section, we provide a formal justification of our approximation based on an interpretation of the original δ quantity, defined by Eq. 3.19, as a probability distribution.

If we denote by $\delta_t(C_t, s_t)$ the right-hand side of Eq. 3.19 without the maximization over C_t, then

$$\begin{aligned} \delta_t(C_t, s_t) &= P(I_t, C_t|s_t) \overbrace{\max_{s_{t-1}} P(s_t|s_{t-1})\, \delta_{t-1}(s_{t-1})}^{\hat{P}(s_t)} \\ &= P(C_t|s_t, I_t)\; \underbrace{P(I_t|s_t) \max_{s_{t-1}} P(s_t|s_{t-1})\, \delta_{t-1}(s_{t-1})}_{\hat{P}(I_t, s_t)}. \end{aligned} \tag{3.44}$$

Although the factor $P(C_t|s_t, I_t)$ is a proper distribution, $\hat{P}(I_t, s_t)$ is not, hence $\delta_t(C_t, s_t)$ cannot be considered as a distribution. However, since our interest is in maximizing over s_{t-1}, it is only the relative proportion of the quantities $\hat{P}(I_t, s_t)$ obtained for different s_t which is important, not their exact values. If we define

$$\hat{P}(s_t|I_t) = \frac{\hat{P}(I_t, s_t)}{\sum_{s'_t} \hat{P}(I_t, s'_t)} \tag{3.45}$$

then Eq. 3.44 can be written as

$$\delta_t(C_t, s_t) \propto \hat{P}(C_t, s_t|I_t) = P(C_t|s_t, I_t)\, \hat{P}(s_t|I_t), \tag{3.46}$$

where $\hat{P}(C_t, s_t|I_t)$ is a proper probability distribution[*]. Considering the original δ formulation (3.19) and the definition of $\delta_t(C_t, s_t)$ (3.44), the equation for the computation of the optimal contour $C_t^*(s_t)$ (3.25) is equivalent to

$$C_t^*(s_t) = \arg\max_{C_t} \delta_t(C_t, s_t), \tag{3.47}$$

which, according to Eq. 3.46, can be written as

$$C_t^*(s_t) = \arg\max_{C_t} \hat{P}(C_t, s_t|I_t). \tag{3.48}$$

Since we want to avoid having to perform a segmentation for each possible action class s_t, we are interested in approximating $\hat{P}(C_t, s_t|I_t)$ by a simpler distribution $Q(C_t, s_t) = Q(C_t)\, Q(s_t)$ where the dependency between C_t and s_t has been dropped. The unique optimal contour for frame I_t can then be obtained by maximizing $Q(C_t)$ over C_t.

Our goal is to find the Q distribution which is as close as possible to the true $\hat{P}$ distribution. A possible way to achieve this is to minimize the Kullback-Leibler (KL) divergence [85] between the two distributions:

$$\mathrm{KL}\big(Q\,||\,\hat{P}\big) = \big\langle \log Q(C_t, s_t)\big\rangle_{Q(C_t,s_t)} - \big\langle \log \hat{P}(C_t, s_t|I_t)\big\rangle_{\hat{Q}(C_t,s_t)} \tag{3.49}$$

where $\langle\cdot\rangle_Q$ denotes the average with respect to Q [†]. After expanding its right-hand side, Eq. 3.49 can be written as

$$\mathrm{KL}\big(Q\,||\,\hat{P}\big) = \big\langle \log Q(C_t)\big\rangle_{Q(C_t)} + \big\langle \log Q(s_t)\big\rangle_{Q(s_t)} - \big\langle \log \hat{P}(I_t, C_t, s_t)\big\rangle_{Q(C_t,s_t)} + \log \hat{P}(I_t). \tag{3.50}$$

Differentiating this with respect to $Q(s_t)$ and $Q(C_t)$ respectively, and equating the result to zero, yields

$$Q(s_t) \propto \exp\left\{\big\langle \log \hat{P}(I_t, C_t, s_t)\big\rangle_{Q(C_t)}\right\}, \tag{3.51}$$

$$Q(C_t) \propto \exp\left\{\big\langle \log \hat{P}(I_t, C_t, s_t)\big\rangle_{Q(s_t)}\right\}. \tag{3.52}$$

---

[*] To facilitate understanding, we intentionally wrote the various quantities involved in a probabilistic fashion—with a hat on top of the P however, to remind ourselves that they are fake distributions. For example, in Eq. 3.45, the denominator corresponds to $\hat{P}(I_t)$.

[†] For instance, for two distributions $P(x)$ and $Q(x)$, with x a continuous variable, we have $\big\langle \log P(x)\big\rangle_{Q(x)} = \int_x Q(x) \log P(x)\, dx$. If x is discrete, the integration is replaced by a summation.

In order to be able to compute $Q(s_t)$, we need to choose a distribution $Q(C_t)$ for which the average in Eq. 3.51 is analytically tractable. Since our interest is in approximating the most likely contour C_t^*, the simplest choice is to use a Dirac distribution centered on the approximate contour $\tilde{C}_t^*$, i.e.,

$$Q(C_t) = \delta(C_t - \tilde{C}_t^*). \tag{3.53}$$

Using this in Eq 3.51 leads to

$$\begin{aligned} Q(s_t) &\propto \hat{P}(I_t, \tilde{C}_t^*, s_t) \\ &\propto P(I_t, \tilde{C}_t^* | s_t)\, \hat{P}(s_t) \\ &\propto P(I_t, \tilde{C}_t^* | s_t) \max_{s_{t-1}} P(s_t | s_{t-1})\, \delta_{t-1}(s_{t-1}). \end{aligned} \tag{3.54}$$

Furthermore, from Eqs. 3.19, 3.44 and 3.46, we have

$$\begin{aligned} \delta_{t-1}(s_{t-1}) &= \max_{C_{t-1}} \delta_{t-1}(C_{t-1}, s_{t-1}) \\ &\propto \max_{C_{t-1}} \hat{P}(C_{t-1}, s_{t-1} | I_{t-1}) \\ &\approx \max_{C_{t-1}} Q(C_{t-1})\, Q(s_{t-1}) \\ &\approx Q(s_{t-1}) \max_{C_{t-1}} Q(C_{t-1}), \end{aligned} \tag{3.55}$$

where we have used the fact that we approximate $\hat{P}(C_{t-1}, s_{t-1} | I_{t-1})$ by $Q(C_{t-1}, s_{t-1}) = Q(C_{t-1})\, Q(s_{t-1})$. Since $\max_{C_{t-1}} Q(C_{t-1})$ is a quantity which does not depend on s_{t-1}, Eq. 3.54 can be written as

$$Q(s_t) \propto P(I_t, \tilde{C}_t^* | s_t) \underbrace{\max_{s_{t-1}} P(s_t | s_{t-1})\, Q(s_{t-1}).}_{w_t(s_t)} \tag{3.56}$$

Since Eq. 3.56 is the same as Eq. 3.26, we conclude that our initial intuitive definition of $\tilde{\delta}_t(s_t)$ can be formally derived from the minimization of the KL divergence between the true distribution $\hat{P}$ and a simpler Q distribution, where the problematic dependency between C_t and s_t has been removed.

In order to compute the right-hand side of Eq. 3.56, we first need to find the approximate most likely contour $\tilde{C}_t^* = \arg\max_{C_t} Q(C_t)$. After expanding the right-hand side of Eq. 3.52 and using Eq. 3.24, we have

$$\begin{aligned} \tilde{C}_t^* &= \arg\max_{C_t} \big\langle \log \hat{P}(I_t, C_t, s_t) \big\rangle_{Q(s_t)} \\ &= \arg\max_{C_t} \Big\langle \log \big(P(I_t, C_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1})\, Q(s_{t-1}) \big) \Big\rangle_{Q(s_t)} \\ &= \arg\max_{C_t} \Big(- E_{\text{image}}(I_t, C_t) + \log P(C_t) + \Big\langle \log \big(P(A_t | s_t)\, w_t(s_t) \big) \Big\rangle_{Q(s_t)} \Big) \end{aligned} \tag{3.57}$$

where $A_t = f_A(I_t, C_t)$ and $w_t(s_t)$ is implicitly defined in Eq. 3.56. Following the notation introduced in Section 3.5, we identify $\log P(C_t)$ as $-E_{\text{contour}}$ and the average as $-E_{\text{prior}}$, hence

$$E_{\text{prior}} = -\Big\langle \log\big(P(A_t|s_t)\, w_t(s_t)\big)\Big\rangle_{Q(s_t)} = -\sum_{s_t} Q(s_t)\, \log\big(P(A_t|s_t)\, w_t(s_t)\big). \tag{3.58}$$

Computing E_{prior} is problematic because the right-hand side of Eq. 3.58 depends on $Q(s_t)$, which itself indirectly depends on E_{prior} through the approximate contour $\tilde{C}_t^*$. A possible solution to this problem could be to alternate between Eqs. 3.56 and 3.58 until convergence is achieved. For example, starting from a random guess of $Q(s_t)$ for each time step t, we can compute the approximate most likely contours $\tilde{C}_t^*$ with Eq. 3.57 and then use Eq. 3.56 to provide a new estimation of $Q(s_t)$. However, this iterative approach is not satisfactory because it requires performing a new image segmentation every time $Q(s_t)$ is re-estimated, and this goes against our main goal of performing a single segmentation at each time step only.

A procedure for computing $\tilde{C}_t^*$ and $Q(s_t)$ in a single step can be derived form the intuition that the action class s_t whose attributes A_t best correspond to those extracted from the image I_t is likely to dominate the average in Eq. 3.58. We could therefore consider $Q(s_t)$ as a free parameter and optimize Eq. 3.58 over $Q(s_t)$. We would then use the resulting $\tilde{C}_t^*$ to compute $Q(s_t)$ using Eq. 3.56. In order to properly optimize Eq. 3.58 over $Q(s_t)$ we need to add the constraints

$$\forall s_t,\ Q(s_t) \geq 0 \quad \text{and} \quad \sum_{s_t} Q(s_t) = 1. \tag{3.59}$$

A possible way to achieve this is to define $Q(s_t = i) = L_i^2$ and to use the method of Lagrange multipliers, which consists in adding a term of the form

$$\beta\Big(1 - \sum_i L_i^2\Big)^2$$

to the right-hand side of Eq. 3.58. With this approach, E_{prior} is given by

$$E_{\text{prior}}(C_t, \mathcal{L}, I_t) = -\sum_i \log\big(P(A_t|s_t = S_i)\, w_t(S_i)\big)\, L_i^2 + \beta\Big(1 - \sum_i L_i^2\Big)^2. \tag{3.60}$$

Since Eq. 3.60 is the same as Eq. 3.37, and the contour $\tilde{C}_t^*$ can be obtained by a single segmentation using the energy (3.35), we conclude that our intuitive competition-between-priors strategy can be formally justified as resulting from the construction of an approximate Q distribution through the minimization of the KL divergence $\mathrm{KL}(Q\,||\hat{P})$.

3.7 On the Benefits of Our Strategy for Competition Among Multiple Priors

In order to emphasize our contribution to the field of variational image segmentation, we would like to present the advantages of using our proposed approach for introducing multiple competing priors into the segmentation formulation, compared to the well-known approach of Cremers et al. [46, 49]. In particular, we will show that the latter introduces an unwanted dependency between the relative contributions of each of the priors, which makes the competition unfair.

As shown in Chapter 2, Cremers et al. propose the following energy for the introduction of multiple shape priors into the segmentation:

$$\begin{aligned} E_{\text{shape}}(\phi, \mathbf{L}) = & \sum_{i=1}^{m-1} \iint_{\Omega} \frac{(\phi - \phi_i)^2}{\sigma_i^2} \chi_i(\mathbf{L})\, dx\, dy + \lambda^2 \iint_{\Omega} \chi_m(\mathbf{L})\, dx\, dy \\ & + \gamma \sum_{i=1}^{n} \iint_{\Omega} |\nabla L_i|\, dx\, dy. \end{aligned} \tag{3.61}$$

Here ϕ is the level-set function of the segmenting contour, ϕ_i are the level-set functions of the prior shapes, σ_i^2 represents the variance of ϕ_i. Furthermore, $\gamma > 0$ and the last energy term imposes smoothness of the labeling function, whereas $\lambda > 0$ and its associated energy term corresponds to a region where no prior is imposed, since the resemblance between ϕ and any of the priors falls bellow a threshold dictated by λ. Moreover, $\mathbf{L} : \Omega \rightarrow \mathbb{R}^n$, $\mathbf{L}(x, y) = (L_1(x, y), \ldots L_n(x, y))$ is a vector-valued labeling function defined over the image domain, which enforces the prior which is most similar to the level set ϕ at each image location. Using this function, the authors employ the $m = 2^n$ vertices of the polytope $[-1, 1]^n$ to encode m different regions, denoted by their respective indicator functions χ_i, $i = 1..m$ (depending on the vector $\mathbf{L}$).

For instance, to encode three priors, a double-valued labeling function is used $\mathbf{L}(x) = (L_1(x), L_2(x))$. The corresponding indicator functions are given by:

$$\begin{aligned} \chi_1(\mathbf{L}) =& \frac{1}{16}(L_1 - 1)^2(L_2 - 1)^2 \\ \chi_2(\mathbf{L}) =& \frac{1}{16}(L_1 + 1)^2(L_2 - 1)^2 \\ \chi_3(\mathbf{L}) =& \frac{1}{16}(L_1 - 1)^2(L_2 + 1)^2 \\ \chi_4(\mathbf{L}) =& \frac{1}{16}(L_1 + 1)^2(L_2 + 1)^2. \end{aligned} \tag{3.62}$$

This means, for instance, that in order to enforce prior 1 at a location (x, y), we need both labels $L_1(x, y)$ and $L_2(x, y)$ to converge to -1 for that location. In a fair competition, we would like to enforce prior 1 if its has the shortest distance (lowest error) with respect to the evolving level-set function at that location $\phi(x, y)$, while also being under the threshold λ, i.e.,

$$\begin{aligned} & \frac{(\phi - \phi_1)^2}{\sigma_1^2} < \frac{(\phi - \phi_2)^2}{\sigma_2^2} \quad \text{and} \\ & \frac{(\phi - \phi_1)^2}{\sigma_1^2} < \frac{(\phi - \phi_3)^2}{\sigma_3^2} \quad \text{and} \\ & \frac{(\phi - \phi_1)^2}{\sigma_1^2} < \lambda^2. \end{aligned} \tag{3.63}$$

To see how the use of the competition method involved in (3.61) might prevent the application of prior 1, in spite of the right conditions being fulfilled (3.63), let us consider the evolution equation for the label component L_1:

$$\begin{aligned}\frac{\partial L_1}{\partial t} = &-\frac{1}{16}\Bigg((L_1-1)(L_2-1)^2\frac{(\phi-\phi_1)^2}{\sigma_1^2} + (L_1+1)(L_2-1)^2\frac{(\phi-\phi_2)^2}{\sigma_2^2} \\ &+ (L_1-1)(L_2+1)^2\frac{(\phi-\phi_3)^2}{\sigma_3^2} + \lambda^2(L_1+1)(L_2+1)^2\Bigg) - \gamma \mathrm{div}\left(\frac{\nabla L_1}{|\nabla L_1|}\right).\end{aligned} \tag{3.64}$$

To simplify understanding, let us now look at the value of $L_1(x,y)$ after the first evolution step, given that we have initialized the labels with the neutral value: $L_1(x,y) = L_2(x,y) = 0,\ \forall (x,y) \in \Omega$:

$$L_1^{t=1} = \frac{1}{16}\left(\frac{(\phi-\phi_1)^2}{\sigma_1^2} - \frac{(\phi-\phi_2)^2}{\sigma_2^2} + \frac{(\phi-\phi_3)^2}{\sigma_3^2} - \lambda^2\right). \tag{3.65}$$

This means that L_1 will advance towards the desired value -1 only if

$$\frac{(\phi-\phi_1)^2}{\sigma_1^2} + \frac{(\phi-\phi_3)^2}{\sigma_3^2} < \frac{(\phi-\phi_2)^2}{\sigma_2^2} + \lambda^2. \tag{3.66}$$

This condition is not necessarily satisfied given (3.63). For instance, it can happen that

$$\frac{(\phi-\phi_3)^2}{\sigma_3^2} > \frac{(\phi-\phi_2)^2}{\sigma_2^2} + \lambda^2. \tag{3.67}$$

Therefore, for reasons which do not depend on the prior 1, given by (3.67), the label L_1 cannot converge towards -1 and thus prior 1 cannot be imposed, even though it would be the fair winner of the competition, according to (3.63).

This is a general problem of the energy formulation (3.61), since it introduces a dependency among the indicator functions corresponding to each prior χ_i, via the shared label components $L_1, L_2, \ldots, L_n$. For instance, in the presented case of three shape priors, it creates an artificial grouping of priors into the ones supporting the convergence of label L_1 towards -1, i.e., priors 1 and 3, and the ones supporting its convergence towards 1, i.e. prior 2 and the threshold λ. Therefore, the evolution of each label component receives mixed influences from several shape priors, as in (3.65). This means that the simple conditions (3.63) for the fair emergence of one winner prior (i.e., the most similar to ϕ) are not sufficient, and supplementary conditions need to be fulfilled (3.66). Such conditions go beyond the terms of fair competition and are therefore unacceptable.

The solution to this problem is to use an individual labeling of the priors, as we propose in (3.37). We use multiple competing priors in a slightly different context: our priors compete globally for one image, and not at the level of image locations. Moreover, the nature of our priors is different, since they are provided by the recognition scheme as functions of the object attribute $P(A_t|s_t = S_i)\, w_t(S_i)$. Nevertheless, the competition concept is similar. As we mentioned, the main difference of our approach is that we use an individual label L_i

for each prior, and thus our "indicator functions", given by L_i^2, do not mix several labels. This solves the problem of mixed influences from several priors in the evolution of one label. As we can see in (3.41), only the prior information $P(A_t|s_t = S_i)\, w_t(S_i)$ is involved in the evolution of L_i. In our approach, the coupling of labels is only at the level of the constraint $(1-\sum_{i=1}^{M} L_i^2)^2 \approx 0$, imposed via the use of the Lagrange multiplier β. Therefore, the label of the prior which best suits the attributes of the detected object will be the one advancing fastest towards value 1, to the detriment of the others. This suits the spirit of fair competition that we intended to implement.

3.8 Learning the Parameters of Our Model

Prior to testing our framework by applying it to the segmentation and behavior recognition of new image sequences, the proposed model needs to be trained. More precisely, its parameters need to be estimated from training data.

Looking at the joint distribution of our model (3.1), we note that its parameters are the ones characterizing the probability distributions $P(s_1)$, $P(s_t|s_{t-1})$, $P(A_t|s_t)$, $P(I_t|A_t, C_t)$ and $P(C_t)$. Supposing that we have at our disposal a training set of N image sequences $\{I^1_{1:T_1}, \ldots, I^N_{1:T_N}\}$ — where T_n is the length of the n-th sequence — the training of our model consists in finding the parameter setting which maximizes the total log-likelihood of the training data, i.e.,

$$\Psi^* = \arg\max_{\Psi} \sum_{n=1}^{N} \log P(I^n_{1:T_n}|\Psi). \tag{3.68}$$

Here Ψ denotes the set of model parameters and

$$P(I_{1:T}|\Psi) = \sum_{s_{1:T}} \int_{A_{1:T}} \int_{C_{1:T}} P(I_{1:T}, A_{1:T}, C_{1:T}, s_{1:T}|\Psi). \tag{3.69}$$

Note that here we write explicitly the dependency on Ψ of the joint distribution defined by (3.1). The summation and integration make the direct optimization difficult because they couple all the factors together.

To simplify the problem, we propose to decompose it in two parts: one corresponding to the HMM which is at the core of our model and the other one corresponding to our segmentation model. To this end, first of all, we suppose that we can directly observe the attributes $A^n_{1:T_n}$ of the training images. This can be realized by the segmentation of the training image sequences. To favor automatic segmentation, the training sequences should contain the object of interest evolving on a simple background, while displaying similar behavior content as the images targeted for recognition in the testing phase. Once the object attributes have been extracted from the training sequences, our problem is reduced to the classical HMM training. In this case, the set of parameters is reduced to the ones characterizing the HMM core of our model, i.e., the parameters of the action class initial and transition distribution $P(s_1)$ and $P(s_t|s_{t-1})$, as well as the parameters of the attribute probability model $P(A_t|s_t)$.

HMM training can be performed either in a supervised or in a unsupervised fashion. In the unsupervised case, the action classes (states) corresponding to the observed attributes are considered as hidden and the parameter estimation can be expressed as:

$$\Psi_H^* = \arg\max_{\Psi_H} \sum_{n=1}^{N} \log P(A_{1:T_n}^n | \Psi_H), \tag{3.70}$$

where Ψ_H denotes the set of HMM parameters and

$$P(A_{1:T}|\Psi_H) = \sum_{s_{1:T}} P(A_{1:T}, s_{1:T}|\Psi_H), \tag{3.71}$$

with

$$\begin{aligned} P(A_{1:T}, s_{1:T}|\Psi_H) &= \prod_{t=1}^{T} P(s_t|s_{t-1}, \Psi_H)\, P(A_t|s_t, \Psi_H), \\ P(s_1|s_0, \Psi_H) &= P(S_1|\Psi_H). \end{aligned} \tag{3.72}$$

This problem can be solved by the Expectation Maximization (EM) algorithm [54], which, for the HMM, yields the Baum-Welch algorithm [11, 120]. The alternative is supervised training, where the action classes corresponding to the training attribute sequences are also considered as visible (observed). To this end, a manual classification of attribute sequences into action classes is necessary. This makes possible the individual estimation of the parameters for each of the probabilities involved ($P(s_1)$, $P(s_t|s_{t-1})$ and $P(A_t|s_t)$) by maximum likelihood. This simplification is due to the fact that by observing the action classes, we can re-write the estimation problem (3.70) as:

$$\Psi_H^* = \arg\max_{\Psi_H} \sum_{n=1}^{N} \log P(A_{1:T_n}^n, s_{1:T_n}^n | \Psi_H). \tag{3.73}$$

Substituting the expression of the HMM joint variable distribution (3.72), we obtain:

$$\Psi_H^* = \arg\max_{\Psi_H} \left(\sum_{n=1}^{N} \log P(s_1^n|\Psi_H) + \sum_{n=1}^{N} \prod_{t=2}^{T_n} \log P(s_t^n|s_{t-1}^n, \Psi_H) + \sum_{n=1}^{N} \prod_{t=1}^{T_n} \log P(A_t^n|s_t^n, \Psi_H) \right), \tag{3.74}$$

which leads to the maximum likelihood estimation, separately for the sets of parameters corresponding to each of the probabilities $P(s_1)$, $P(s_t|s_{t-1})$ and $P(A_t|s_t)$. In particular, for the initial action class distribution $P(s_1)$, this estimation yields, for each action class $s_1 \in S$, its relative frequency of occurrence at the first frame of the sequences from the given training set. Similarly, for the transition probability distribution $P(s_t|s_{t-1})$, the estimation yields, for each action class pair (s_t, s_{t-1}), its relative frequency of occurrence among the consecutive frames of the sequences from the given training set. The supervised training method of the HMM is potentially more reliable than the unsupervised one — which relies on an automatic optimization algorithm susceptible to local minima — but also more time consuming for the human operator, due to the necessary manual labeling of the attribute sequences.

Let us now look at the training of the segmentation model parameters, i.e., the parameters of $P(C_t)$ and $P(I_t|A_t, C_t)$, the latter being actually the parameters of $E_{\text{image}}(I_t, C_t)$, due to (3.9). As shown in Chapter 2, an example of parameters for the image-dependent segmentation energy $E_{\text{image}}(I_t, C_t)$, is given by the intensity means corresponding to the object and background region, respectively. Such parameters can be learned from training data by maximum likelihood estimation, given appropriate segmentations of training image sequences. The learning of the parameter values for $E_{\text{image}}(I_t, C_t)$ and $P(C_t)$ imposes some degree of similarity (in terms of these parameters) among the images of a test sequence — since the model is fixed throughout the test sequence — and also between the images of the testing set and the ones of the training set. Some relief from this constraint would be brought by learning these parameters from the first frame of a test sequence, assuming that they remain relatively constant throughout the test sequence. The least engaging option, that we also chose in our implementations in Chapter 4, is to deduce and update these parameters dynamically at testing time, during the segmentation of each image. In this case, there is no need for image similarity between the training and the testing set. An example of such a dynamic estimation is given by the Chan-Vese segmentation model, presented in Chapter 2.

3.9 Extension of Our Framework for the Recognition of a Predefined Behavior Set

In simple application cases, where the analyzed behaviors are composed of few and relatively well differentiated action classes, whose succession can be well characterized by the set the class transition probabilities (3.3), the use of our model represented by the DBN in Fig. 3.4 is sufficient to model behavior dynamics and perform inference about object behavior, as described in the previous sections. One example of such application, together with a proposed solution, using an implementation of our framework as depicted in Fig 3.4, is detailed in Chapter 4. Nevertheless, many practical applications require analysis of complex behavior scenarios, involving numerous classes, often poorly discriminated in terms of the available attributes. In such cases, the behavior inference process can be greatly aided by imposing stricter coherence conditions on the resulting succession of behavior classes, stemming from prior knowledge about possible behaviors. In particular, in the following we will consider that the analyzed behaviors can be assigned to one of a finite set of behavior types. We will focus on the case of isolated behavior recognition, by considering finite length image sequences, featuring one of the set of predefined behavior types. A behavior of a certain type can be characterized by its particular decomposition into a succession of basic actions, each belonging to an action class. Since in most applications there exist basic actions which are common among different behaviors, the set of basic actions is modeled as shared among behaviors of all types. This means that a behavior of any type can potentially include any action class, its composition being given by the values of the model parameters.

In order to capture these considerations, we modify the DBN in Fig. 3.4 to yield the

DBN in Fig. 3.5. The modification consists in introducing a dependence between the action class at each moment s_t and the exhibited behavior type b, which belongs to a finite set of behavior types $B = \{B_1, B_2, \dots B_K\}$. Since all the action classes depend on a single behavior type, the model also states that the analyzed image sequences feature a single behavior type, i.e., that we perform isolated behavior recognition.

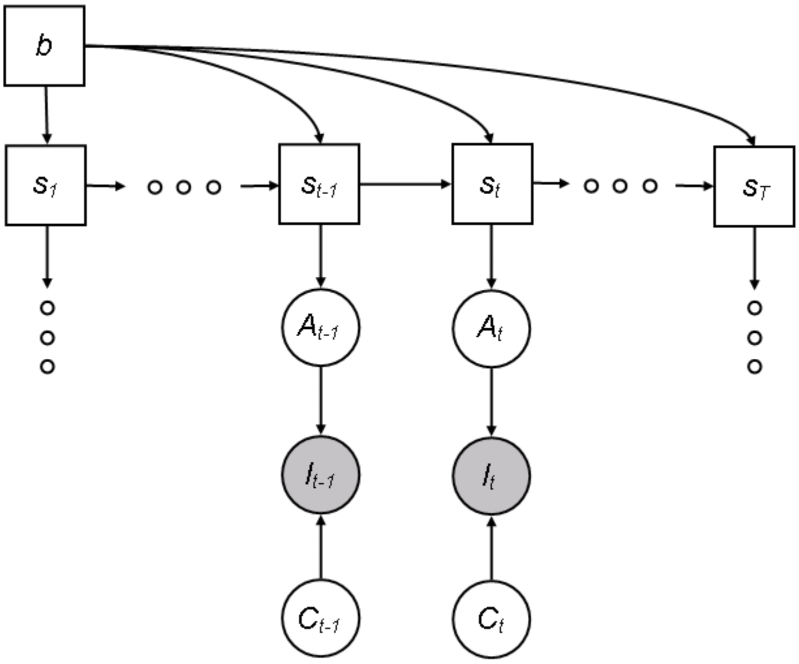


Figure 3.5 — Our DBN for the recognition of a predefined behavior set. Compared to our original model, note the added dependence of the action class at each moment s_t on the particular behavior type b which is being exhibited. Additionally, please observe that the recognized sequence is considered to feature one type of behavior, meaning that we perform isolated behavior recognition. We maintain the same convention of representing hidden variables by clear nodes, observed variables by shaded nodes, discrete variables by square nodes and continuous variables by circular nodes.

Let us now look at how this modification affects the formulation of our joint segmentation / behavior recognition problem. We will present the joint variable distribution of our new model, detail the estimation of our unknowns through modified Viterbi decoding and variational segmentation and then we will explain model training. To this end, we will follow the same rationalism as in case of our initial model of Fig. 3.4 (presented in Sections 3.3, 3.4 and 3.8), while emphasizing the differences due to the model modification.

The joint variable distribution which characterizes our model represented in Fig. 3.5 can be written as:

$$P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}, b) = \prod_{t=1}^{T} P(I_t|A_t, C_t)\, P(C_t)\, P(A_t|s_t)\, P(s_t|s_{t-1}, b)\, P(b), \qquad (3.75)$$

with $P(s_1|s_0, b) \equiv P(s_1|b)$ — the initial action class distribution given the behavior type b. The modifications to this distribution with respect to (3.1) consist of the added dependence of the action class initial and transition distributions $P(s_1|b)$ and $P(s_t|s_{t-1}, b)$ on the behavior type b, as well as the addition of the prior probability of the behavior type $P(b)$. The latter can be chosen depending on the application, in order to reflect the fact that some behavior types may be more probable than others. In the absence of such information, a uniform prior $P(b)$ can be chosen. Given its dependence on the behavior type, the transition distribution $P(s_t|s_{t-1}, b)$ is characterized by K sets of transition probabilities $T^k = \{t^k_{ij}\},\ k = 1..K$:

$$P(s_t = S_j|s_{t-1} = S_i, b = B_k) = t^k_{ij}, \quad i, j = 1..M, \tag{3.76}$$

under the same constraints (3.4) applied to each set T^k. Similarly, the initial action class distribution is given by $\pi^k = \{\pi^k_i\},\ k = 1..K$, with

$$P(s_1 = S_i|b = B_k) = \pi^k_i, \quad i = 1..M. \tag{3.77}$$

Regarding the rest of the probabilities composing the distribution (3.75), i.e., $P(I_t|A_t, C_t)$, $P(C_t)$ and $P(A_t|s_t)$, the considerations expressed in Section 3.3 remain valid.

Our joint segmentation / behavior recognition problem in terms of the modified model implies the addition of the optimization with respect to the behavior type to the probabilistic formulation (3.12), yielding:

$$(b^*, s^*_{1:T}, C^*_{1:T}) = \arg\max_{\substack{b\\ s_{1:T}\\ C_{1:T}}} P(I_{1:T}, C_{1:T}, s_{1:T}, b). \tag{3.78}$$

Considering the estimation of the likelihood of the most likely configuration $(b^*, s^*_{1:T}, C^*_{1:T})$, we can write:

$$\begin{aligned} P(I_{1:T}, C^*_{1:T}, s^*_{1:T}, b^*) &= \max_{\substack{b\\ s_{1:T}\\ C_{1:T}}} P(I_{1:T}, C_{1:T}, s_{1:T}, b) \\ &= \max_{b} \max_{\substack{s_{1:T}\\ C_{1:T}}} P(I_{1:T}, C_{1:T}, s_{1:T}, b). \end{aligned} \tag{3.79}$$

The last equality suggests the use of a Viterbi decoding approach similar to the one used for our original model in order to compute the inner maximization for each behavior type $b \in B$, followed by the maximization over the behavior type of the resulting quantities. The Viterbi decoding algorithm can be deduced in a similar manner to the one used for our original model, described in Section 3.4.

Analogously to Section 3.4, we define a δ quantity for time slice t, action class s_t and behavior type b as:

$$\delta_t(s_t, b) = \max_{\substack{s_{1:t-1}\\ C_{1:t}}} P(I_{1:T}, C_{1:T}, s_{1:T}, b). \tag{3.80}$$

Its recursive computation is given by:

$$\delta_t(s_t, b) = \max_{C_t} P(I_t, C_t|s_t) \max_{s_{t-1}} P(s_t|s_{t-1}, b)\, \delta_{t-1}(s_{t-1}, b). \tag{3.81}$$

Following the retrieval of the optimal action class sequence $s^*_{1:T}$ by backtracking from the maximizing arguments of Eq. 3.81, the optimal contour sequence $C^*_{1:T}$ would be retrieved as:

$$C^*_t = \arg\max_{C_t} P(I_t, C_t | s^*_t). \tag{3.82}$$

Similarly to our original model, the recursion for the computation of δ implies the estimation via image segmentation of the locally most likely contour $C^*(s_t)$ corresponding to each action class $s_t \in S$:

$$C^*_t(s_t) = \arg\max_{C_t} P(I_t, C_t | s_t), \tag{3.83}$$

where $P(I_t, C_t|s_t)$ is given by Eq. 3.24. In similar fashion to our original model, we wish to avoid the computational costs associated with the repeated segmentation procedure and thus replace the estimation of $\delta_t(s_t, b)$ by (3.81) with:

$$\tilde{\delta}_t(s_t, b) = P(I_t, \tilde{C}^*_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}, b)\, \tilde{\delta}_{t-1}(s_{t-1}, b), \tag{3.84}$$

equivalent to

$$\tilde{\delta}_t(s_t, b) = P(I_t, \tilde{C}^*_t | s_t)\, w_t(s_t, b), \tag{3.85}$$

where $w_t(s_t, b)$ is defined as

$$w_t(s_t, b) = \max_{s_{t-1}} P(s_t | s_{t-1}, b)\, \tilde{\delta}_{t-1}(s_{t-1}, b). \tag{3.86}$$

Here, the contour $\tilde{C}^*_t$ is an approximation of the optimal contour C^*_t from Eq. 3.82, obtained by a single segmentation of image I_t, according to:

$$\tilde{C}^*_t = \arg\max_{C_t} \left(\max_{s_t, b} P(I_t, C_t | s_t)\, w_t(s_t, b) \right). \tag{3.87}$$

The difference with respect to the formulation in our original model (3.29) is given by the added maximization with respect to the behavior type b. Defining

$$\tilde{w}_t(s_t) = \max_{b} w_t(s_t, b), \tag{3.88}$$

we can express (3.87) as:

$$\tilde{C}^*_t = \arg\max_{C_t} \left(\max_{s_t} P(I_t, C_t | s_t)\, \tilde{w}_t(s_t) \right). \tag{3.89}$$

which is similar to the expression obtained for our initial model in Section 3.4. Substituting $P(I_t, C_t|s_t)$, Eq. 3.89 becomes:

$$\tilde{C}^*_t = \arg\max_{C_t} \left(e^{-E_{\text{image}}(I_t, C_t)}\, P(C_t)\, \max_{s_t} P(A_t = f_A(I_t, C_t) | s_t)\, \tilde{w}_t(s_t) \right). \tag{3.90}$$

For segmentation, we use the same variational segmentation formulation as for our initial model, described in Section 3.5. The difference lies in the attainment of the infused a priori knowledge, this time given by $P(A_t|s_t)\, \tilde{w}_t(s_t)$, with $A_t = f_A(I_t, C_t)$.

Returning to the Viterbi decoding scheme, we associate the following initialization to our recursion for $\tilde{\delta}_t(s_t, b)$ (3.84):

$$\tilde{\delta}_1(s_1, b) = P(I_1, \tilde{C}_1^* | s_1)\, P(s_1|b)\, P(b). \tag{3.91}$$

The contour C_1^* is obtained by the segmentation of the first image I_1, following the same considerations as in Section 3.4.

Similarly to our Viterbi decoding formulation in Section 3.4, for the purpose of retrieving the optimal action class sequence corresponding to the optimum behavior type, we store the argument which satisfies the maximization from the computation of $\tilde{\delta}_t(s_t, b)$ (3.84), for each time slice $t > 1$, each value of s_t and each value of b:

$$\psi_t(s_t, b) = \arg\max_{s_{t-1}} P(s_t|s_{t-1}, b)\, \tilde{\delta}_{t-1}(s_{t-1}, b), \quad s_t \in S,\ b \in B. \tag{3.92}$$

Therefore, at the final time moment T, we can retrieve the optimal (winning) behavior type, as:

$$b^* = \arg\max_{b} \max_{s_T} \tilde{\delta}_T(s_T, b). \tag{3.93}$$

Its corresponding optimal action class sequence $s_{1:T}^*$ can be retrieved by backtracking:

$$\begin{aligned} s_T^* &= \arg\max_{s_T} \tilde{\delta}_t(s_T, b^*), \\ s_t^* &= \psi_{t+1}(s_{t+1}^*, b^*), \qquad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{3.94}$$

Concerning the training of our modified model from Fig. 3.5, we note that this model has a similar structure and set of parameters as our initial model in Fig. 3.4, with the exception of the action class dependence on the behavior type, implying a difference in action class transition and initial probabilities. Therefore, the considerations referring to the training of our initial model, presented in Section 3.8, can also be applied to our modified model from Fig. 3.5. The main difference regards the training of the action class initial and transition distributions $P(s_1|b)$ and $P(s_t|s_{t-1}, b)$, respectively, which will now be realized separately for each of the behavior types involved $b \in B$, from specific training data.

3.10 Summary

To sum up, we present a schematic description of the steps involved in the use of our framework for joint segmentation and behavior recognition. We unify the descriptions corresponding to the two proposed models, denoted as "model A" (Fig. 3.4) and "model B" (Fig. 3.5), respectively. The use of our framework consists of the following:

- **Training phase:** estimate parameters of the model (A or B) from training attribute sequences, as explained in Sections 3.8 and 3.9, respectively.
- **Testing phase:** perform joint segmentation and behavior recognition on new image sequences $I_{1:T}$:

1. Segment first image in the sequence I_1, according to the options given in Section 3.4, resulting in contour $\tilde{C}_1^*$.
2. Initialize δ variables according to (3.30), for model A and (3.91) for model B.
3. For t = 2..T
   - Compute $w_t(s_t)$, $s_t \in S$ according to (3.28) for model A or $\tilde{w}_t(s_t)$, $s_t \in S$ according to (3.88) and (3.86) for model B.
   - Estimate contour $\tilde{C}_t^*$ by segmenting image I_t using energy (3.35), where the priors are given by $P(A_t|s_t)\, w_t(s_t)$ for model A and by $P(A_t|s_t)\, \tilde{w}_t(s_t)$ for model B, with $A_t = f_A(I_t, C_t)$. The initial contour for the segmentation is given by $\tilde{C}_{t-1}^*$.
   - Compute $\delta_t(s_t)$ and $\psi_t(s_t)$, $s_t \in S$, using (3.26) and (3.31) for model A, or compute $\delta_t(s_t, b)$ and $\psi_t(s_t, b)$, $s_t \in S$, $b \in B$, using (3.84) and (3.92) for model B.
4. For model B, estimate optimal behavior type using (3.93).
5. Backtrack to infer the action class of each image $s_{1..T}^*$ using (3.32) for model A and (3.94) for model B.

3.11 Conclusion

In this chapter, we presented a general framework for the cooperative resolution of the tasks of image segmentation and behavior recognition from image sequences. The cooperation of the two tasks enables the sharing of all existing information resources, which is beneficial to both tasks. On the one hand, dynamical probabilistic priors are offered by the recognition process to guide the segmentation of each image. On the other hand, accurate segmentation results enable the extraction of the desired attributes that are used by the recognition process.

More specifically, our framework is based on the formulation of a Dynamic Bayesian Network, which unites the estimation of the object contour in each image with the classification of the corresponding extracted attributes. The DBN enables the statement of our joint estimation problem in terms of probabilistic inference, thereby allowing its resolution via a variation of the Viterbi decoding algorithm for HMMs. The resulting strategy consists of interleaving a dynamic programming scheme for action class estimation with the greedy estimation of the optimal contour for each image by variational segmentation. In this context, we propose a novel variational segmentation formulation, which combines image-related constraints on the contour with multiple priors over the attributes of the detected object, offered by the recognition scheme. The integration of multiple priors into the segmentation is performed via a competition approach, which can be considered the equivalent of the maximization of a probabilistic criterion with respect to the class label, as required by the probabilistic inference formulation. We also explain how the model parameters can be learned from appropriate training data.

An advantage of formulating our framework in terms of a DBN is that it allows the explicit statement of the assumptions that we make, regarding the dependencies among the different variables involved. In particular, this facilitates the understanding of the model and of the possibilities for its extension / modification, depending on the applications that need to be solved. One such modification, that we treat in this chapter, concerns the ability of the model to incorporate constraints regarding the possible exhibited behaviors. More precisely, the behaviors are restricted to a set of allowed behavior types, each characterized by a particular succession of basic actions. We show that the training and inference strategies for our original model can be applied with slight modifications to its modified version.

To conclude, our work constitutes an important contribution to the field of behavior recognition, by reconsidering the general assumption that attribute extraction and actual behavior recognition are two separate tasks, solved sequentially. In our framework, the collaboration of the two tasks allows the maximum exploitation of the available information, towards improved results for both tasks. Moreover, we introduce a novel approach for variational image segmentation, by incorporating multiple dynamic attribute priors provided by the collaborating recognition process.

Our general framework for joint segmentation and behavior recognition presents several degrees of freedom, which allow its flexible adaptation to the particular needs of various applications. This freedom mainly concerns the choice of attributes employed in the recognition process, the choice of the attribute probability model for the involved action classes and finally the choice of the image-driven segmentation energy, according to the particular type of images considered. Additional options regard the training of the DBN parameters and the segmentation of the first image of each test sequence. In the next chapter, we will present specific examples for each of the free parameters of our framework, and illustrate the use of the resulting models in the resolution of two particular applications, thereby demonstrating the efficiency of our general framework.

4

Applications and Solutions Implementing our Framework

In this chapter we present two applications pertaining to the field of gesture recognition, together with their solutions, derived from our general framework for cooperative segmentation and behavior recognition, introduced in Chapter 3. The first application regards finger-counting recognition and can be assigned to the area of human-computer interaction. We solve it using the framework that we proposed in Section 3.3 of Chapter 3. To this end, we instantiate the free parameters of our framework, by choosing specific probability and image-based segmentation models. These consist of a Gaussian probability model in the level set function space and of the piecewise-constant Chan-Vese segmentation model, respectively. We explain how we estimate the values of model parameters from the available training data. Finally, we present experimental results of segmentation and gesture recognition, obtained by applying our instantiated framework on difficult image sequences, featuring a cluttered background, as well as noise and occlusions of the target object.

The second application concerns finger-spelling recognition, which is a research topic in the field of sign-language recognition. Its resolution is based on the extension of our framework for the recognition of a predefined behavior set, introduced in Section 3.9 of Chapter 3. Following the same course as for our first application, we begin by choosing particular probabilistic and image-based segmentation models. This time, our probabilistic model for each class relies on a distance function between the evolving contour and a PCA-based prior contour corresponding to the class. We detail the estimation of model parameters from our training database. We finish off by presenting segmentation and recognition results, obtained by testing our model on difficult image sequences, which capture a hearing-impaired person gesturing in realistic conditions that involve a cluttered background.

The material in this chapter is based on work that we have published in [72–75].

4.1 Introduction

Apart from spoken language, hand gestures are a natural part of human communication, serving to express emotion and intention. Moreover, for the hearing-impaired, gestures represent the main means of communication. In this context, the computer vision community has devoted considerable effort to the recognition of human gestures. The automatic visual recognition of gestures is an issue of great practical impact in applications such as teleconferencing, sign-language translation or advanced interfaces for human-computer interaction. Regarding the latter application, gestures are a more natural way of interacting with a computer than manipulating the mouse or keyboard, since they are already an established means of human communication. Based on these considerations, we will focus our attention on two gesture recognition applications, that we will solve using our cooperative framework for image segmentation and behavior recognition, introduced in Chapter 3.

The first application that we consider belongs to the area of human-computer interaction (HCI) and regards the recognition of finger-counting gestures. Such gestures could be used in a variety of scenarios involving the transmission of different commands to the computer (choice of menus, choice of operation sequences, etc). Related work has been presented in [94], where a hand gesture recognition method is proposed, as a part of a "stereo active vision interface" system. Each gesture consists of showing a different number of fingers to the camera. Towards recognition, the system undergoes a phase of hand detection, followed by the tracking of the hand region as a skin-colored blob. Actual recognition relies on empirical measures over the hand skeleton, extracted from the binarized hand region. The hand is supposed to be roughly in vertical orientation. Recognition also involves the assumption that the correct hand region has been extracted by the tracking module. In [96], gesture recognition for the control of a video-game is performed via three steps: hand segmentation (based on skin color), hand tracking (using a constant velocity model) and identification of the hand configuration from a set of extracted features, using a finite state model. The gesture set is composed of four hand gestures, out of which one has four subclasses, differing in hand orientation. In [144], a game based on HCI involves the recognition of three hand gestures, differing in finger number. The recognition algorithm consists of three stages: skin-color detection, time-delimitation of each gesture (by studying hand-center motion) and actual recognition, relying on the count of finger regions, subsequent to their extraction by morphological operations. Literature reviews concerning visual-based gesture recognition can be found in [56, 117, 158], with [117] focusing on gesture recognition for HCI.

In all these reviews, as well as in the above-mentioned papers, the recognition approaches are divided into an image analysis phase, resulting in the extraction of a set of image features, and a subsequent recognition phase, where gestures are recognized based on the extracted features, and on knowledge encapsulated in a particular gesture model. Therefore, the recognition phase is heavily dependent on the result of analysis phase (and extracted features), without any possibility to intervene. In contrast to these approaches, for our finger-counting application we use a framework where the two phases are performed in collaboration, allowing the improvement of final results by the sharing between them of the maximum amount of available information. This framework, which has been developed in Chapter 3, is instantiated in this chapter with a particular image-based segmentation energy, as well as an attribute probability model capturing the specifics of each gesture class. The parameters of the resulting concrete model are then learned from training data. Model testing on difficult situations including noisy images, occlusions of the gesturing hand and a cluttered background yields promising segmentation and recognition results. Additionally, our model does not impose restrictions on the position, scale or orientation of the hand, unlike some of the existing methods, since it naturally incorporates the alignment of the prior model to the hand contour into the segmentation formulation.

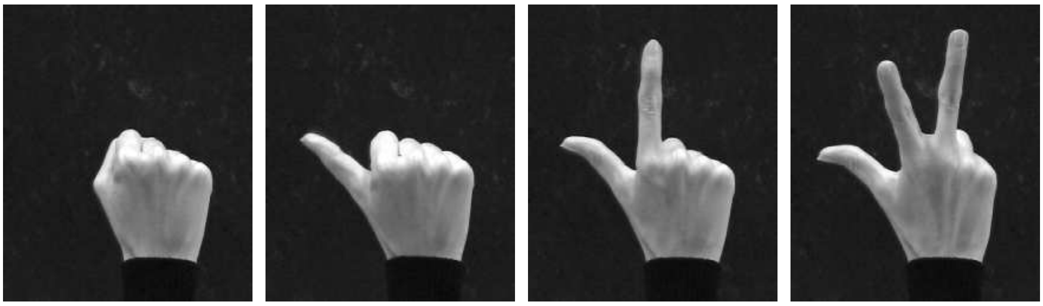
The second application that we address concerns finger-spelling recognition, which pertains to the larger field of sign-language recognition. Sign language is the main means of communication within hearing-impaired communities. It is a visual language, in which the signer conveys meaning through dynamic combinations of hand shapes, arm and body motions and orientations, as well as facial expressions. Information is mainly conveyed through a word-level sign vocabulary. Finger-spelling is the component of sign language which acts as bridge with the surrounding (oral) languages. It consists of manual representations of alphabet letters [109] and it is used for spelling words that have no sign equivalent (such as proper nouns or foreign words), when teaching/learning a sign language or for clarification purposes.

For word-level sign recognition, the most successful approaches [151, 153] rely on the use of devices such as data-gloves and magnetic trackers to extract hand shapes and motions. Compared to these, purely vision-based approaches are preferable, since they are cheaper in terms of technology and also less cumbersome for the signer. Among these, the American Sign Language (ASL) recognition system proposed in [139] tracks hands as skin-colored blobs, extracts global features (such as positions, inertia axis angles and bounding ellipse eccentricities), and then classifies them via Hidden Markov Models (HMMs). In [19], linguistic high level descriptions of the hands' motion, shape and relative positions are extracted from video sequences. These are filtered using Independent Component Analysis (ICA) and classified using a bank of Markov models trained for individual signs. An approach for Australian sign language recognition, based on similar ideas is presented in [77]. It tracks the hands and face of the signer based on skin color and extracts a set of geometric features (positions, global shape descriptors and movement directions), which are then used as input into a HMM classifier. All these systems have shown good performances in their respective sign recognition tasks, but they cannot be directly applied to finger-spelling recognition because of the different nature of the problem, as shown in the following.

Generally, in finger-spelling the discrimination between letters is based on particular hand and finger configurations, rather than on global hand and arm motions, as in word-level signing. Thus, global features (such as the ones used in the above-mentioned systems) are not adequate for finger-spelling recognition and one needs to employ more detailed descriptions of the hand shape. In [91], finger-spelling recognition is addressed by the extraction of hand masks based on skin color and subsequent classification via nearest neighbor template matching and deterministic boosting. In [69], the authors present a recognizer for Australian finger-spelling (which uses two handed dynamic signs, unlike ASL). They obtained good recognition results based on general geometric and motion features, recognized using HMMs. In both approaches, the feature extraction phase relies on skin color for hand region detection and performance is only guaranteed in a controlled laboratory environment, with constant-color background and similar lighting conditions throughout the training and testing phases. In [61], isolated ASL finger-spelling poses are recognized using a special capture setup with a multi-flash camera, which helps detect depth discontinuities within the scene, while assuming constant background. The method helps disambiguate cases of hand self-occlusion, at the cost of using specialized equipment, and still requires a plain, constant background.

Similarly to the cited gesture recognition approaches related to our finger-counting application, all the above-presented sign-language and finger-spelling recognition methods strongly depend on the feature extraction task, fact which could compromise their performance. That is why they are obliged to impose restricting conditions on the signing environment (background, lighting), and often on the hand position, size and orientation (assumed to remain constant and similar to training values). In contrast to these methods, we introduce a method for finger-spelling recognition which is robust against cluttered background and changing lighting conditions, while being invariant to 2D similarity transformations of the signing hand (translation, rotation and scaling). This is achieved by performing feature extraction and classification / recognition jointly, rather than separately, as is done in the existing recognition methods. To this end, we employ the extension of our collaborative segmentation / recognition framework designed for the recognition of a predefined behavior set, introduced in the Section 3.9 of Chapter 3. In this chapter, we propose a concrete implementation of our framework, derived by using a particular image-based segmentation energy, and a PCA-derived attribute probability model for each action class. The latter is based on a PCA-represented prior contour corresponding to its action class, which evolves dynamically during segmentation alongside the main contour, adapting to new image content. To improve the robustness of segmentation, as well as to reduce computation costs, we introduce a pruning strategy, allowing us to select a reduced number of active priors for each image in the sequence, based on their probabilities given by the recognition process. By the infusion of supplementary knowledge gained by recognition from training and from previously analyzed images, our segmentation process is able to cope with difficult situations of cluttered background, without imposing conditions on the lighting conditions. Moreover, invariance with respect to 2D similarity transformations is obtained by including the alignment of the prior models to the current hand shape into the segmentation formulation.

The remainder of this chapter is organized as follows. Section 4.2 presents the finger-counting application and Section 4.3 deals with the finger-spelling application. Both sections include a description of the task, followed by the introduction of the segmentation and probabilistic models used to instantiate the respective frameworks. Next, model training is explained and finally experimental results are presented. Section 4.4 concludes the chapter.



Class 0 Class 1 Class 2 Class 3

Figure 4.1 — Samples from the four gesture classes that we use in our finger-counting application.

4.2 A Finger-counting Application

4.2.1 Task Description

In our finger-counting application, we identify four gesture classes consisting of a right hand (facing the camera) going through four finger configurations: fist (Class 0), thumb extended (Class 1), thumb and index finger extended (Class 2) and thumb, index, and middle finger extended (Class 3). An example image of each gesture class is shown in Fig. 4.1.

Our typical gesture image sequences depict finger-counting from 1 to 3 (starting from the fist position) and from 3 to 1 (ending with the fist position), which is, in terms of gesture class successions, 0,1,2,3 and 3,2,1,0. Our aim is to perform joint segmentation and behavior recognition of image sequences containing such successions; i.e., for each image, extract the segmenting contour of the hand and determine the gesture class to which it belongs.

To solve our application, we use our general framework described in Sections 3.3 and 3.5 of Chapter 3. We start by instantiating this framework with particular image-based segmentation and attribute probability models. Then, we estimate the model parameters from training data and finally we test the resulting implementation on new gesture image sequences.

4.2.2 Implementation Based on Our Framework

Our application features a relatively small number of gesture classes (four), which are quite well discriminated in terms of the hand contour. Therefore, the object attribute that we employ is the level set function corresponding to the hand contour $f_A(I_t, C_t) = \phi(I_t, C_t)$ (for better readability, in the following $\phi(I_t, C_t)$ will be expressed simply as ϕ). The level set function is defined as $\phi : \Omega \to \mathbb{R}$, where Ω is the image domain. In particular, ϕ is chosen to be the signed distance function to the contour:

$$\begin{cases} \phi(x,y) = d(x,y), & \text{for } (x,y) \in \omega, \\ \phi(x,y) = -d(x,y), & \text{for } (x,y) \in \Omega \setminus \omega, \\ \phi(x,y) = 0, & \text{for } (x,y) \in C, \end{cases} \tag{4.1}$$

where $\omega \subset \Omega$ denotes the region enclosed by the contour C and $d(x, y)$ represents the Euclidian distance from point (x, y) to the contour C. Thus, the contour C_t is given by the zero level set of function ϕ: $C \equiv \{(x, y) : \phi(x, y) = 0\}$. We will therefore express the probabilistic and segmentation models of our framework, corresponding to each time slice t, in terms of the function ϕ, which represents the contour at time slice t.

For the image- and contour-based terms in the segmentation energy (3.35), we use the piecewise-constant Chan-Vese model [30], presented in Chapter 2:

$$\begin{aligned} E_{\text{image}}(I_t, \phi) + \nu E_{\text{contour}}(\phi) = &\iint_\Omega (I_t - \mu_+)^2 H(\phi)\,dx\,dy + \iint_\Omega (I_t - \mu_-)^2 H(-\phi)\,dx\,dy \\ &+ \nu \iint_\Omega |\nabla H(\phi)|\,dx\,dy. \end{aligned} \tag{4.2}$$

Here I_t represents the gray-level value of an image location (for better readability the index (x, y) was dropped from the notation). H is the Heaviside function $H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0, \end{cases}$ and μ_+, μ_- are the mean image intensities corresponding to the positive, respectively negative regions of ϕ (i.e., the inside, respectively outside, of the hand region). This term aims to separate the two regions (background/hand) by maximizing the distance between their observed mean intensities.

To describe each gesture class S_i, we use a local Gaussian model of the level set function [126]:

$$p_i^{(x,y)}(\phi) = p^{(x,y)}(\phi|\phi_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i(x, y)} e^{-\dfrac{(\phi(x, y) - \phi_i(x, y))^2}{2\sigma_i^2(x, y)}}. \tag{4.3}$$

Here $(x, y) \in \Omega$ is an image location, ϕ_i represents the average level set function of class S_i and the variance $\sigma_i(x, y)$ models the local variability of the level set at (x, y). The two parameters ϕ_i and σ_i are estimated from appropriate training data for each gesture class S_i. Assuming the independence of ϕ values across image locations, the probability density function of the level set function ϕ, corresponding to class S_i, is given by the product of local ϕ probabilities over the image domain:

$$P_i(\phi) = P(\phi|s_t = S_i) = \prod_{(x,y)\in\Omega} \left(p_i^{(x,y)}(\phi)\right)^{dx\,dy}, \tag{4.4}$$

where $dx\,dy$ represents the infinitesimal bin size.

For a “fair” evaluation of the probability of a level set function ϕ with respect to a gesture class model S_i, represented by the mean and variance parameters (ϕ_i and σ_i), we need to align the model with respect to ϕ prior to probability evaluation. We opt for an alignment with respect to similarity transformations, including translation, rotation, and scaling. A viable alternative, involving more parameters, would be the alignment with respect to affine transformations. We denote as $h_{\boldsymbol{\tau}^i}$ the similarity transformation corresponding to the alignment of class S_i. The parameters of this transformation are $\boldsymbol{\tau} = \{s, \theta, T_x, T_y\}$, i.e., the scale, rotation angle and x- and y-axis translations, respectively. Their values are re-estimated at each segmentation step via gradient descent. The similarity transformation acts on the (x, y) coordinates of the model parameters ϕ_i and σ_i:

$$h_{\boldsymbol{\tau}}\left([x \;\; y]^\top\right) = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \tag{4.5}$$

Within the prior energy (3.37), we substitute the probabilities $P(A_t|s_t = S_i)$ with $P_i(\phi)$ from (4.4) and augment by similarity transformations $h_{\boldsymbol{\tau}^i}$ (4.5) that align the prior model of class S_i with contour ϕ, yielding:

$$\begin{aligned} E_{\text{prior}}(\phi, \mathcal{L}, \boldsymbol{\tau}^{i=1..M}) = \sum_{i=1}^{M} \Bigg(-\log w_t(S_i) + \iint_\Omega \Bigg(\frac{(\phi(x,y) - \phi_i(h_{\boldsymbol{\tau}^i}(x,y))/s^i)^2}{2\sigma_i^2(h_{\boldsymbol{\tau}^i}(x,y))} \\ + \log \sigma_i(h_{\boldsymbol{\tau}^i}(x,y)) \Bigg) dxdy \Bigg) L_i^2 + \beta \left(1 - \sum_{i=1}^{M} L_i^2 \right)^2, \end{aligned} \tag{4.6}$$

where $w_t(S_i)$ is the relative confidence of class S_i (3.28), as explained in Chapter 3.

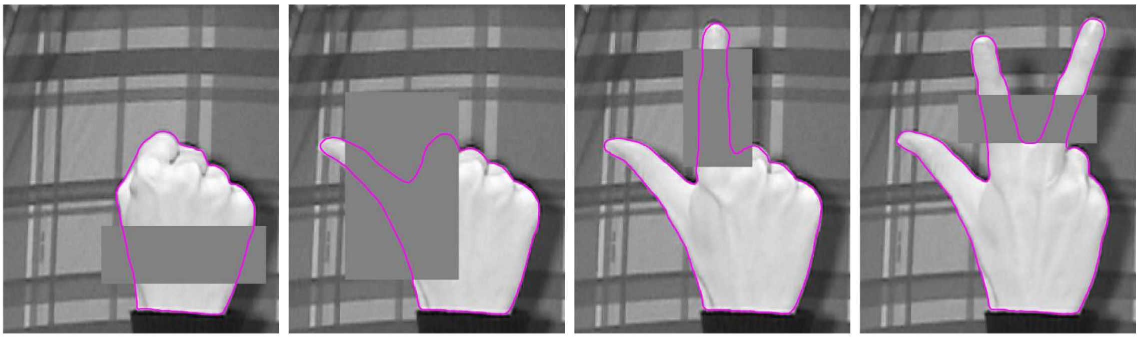
The total energy (3.35), combining the image-based term (4.2) and the prior term (4.6), is minimized via the calculus of variations and gradient descent, following (3.39), (3.40) and (3.41). This results into evolution equations for the contour ϕ, the labels $\mathcal{L}$ and the alignment parameters $\boldsymbol{\tau}^{i=1..M}$, which are presented in the Section A.2 from the Appendix of this thesis. The numerical approximation of these equations is also described in the same section of the Appendix.

4.2.3 Training the Model

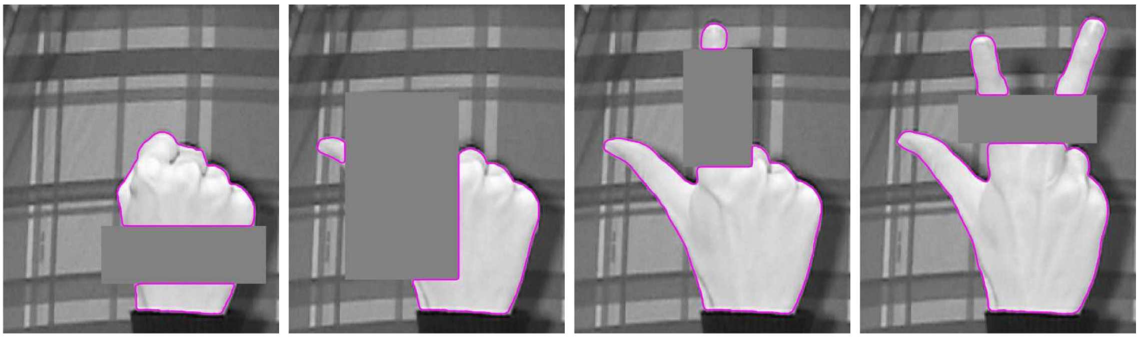
In the training phase, we use counting gesture sequences (0,1,2,3 and 3,2,1,0) performed on a simple contrasting background, as in Fig. 4.1. We begin by segmenting the gesturing hand in each of these images. The good discrimination of the hand grey-level with respect to the uniform dark background allows us to use variational segmentation with the piecewise-constant Chan-Vese model (the image-based term (4.2)). Then, we manually assign gesture class labels to the segmentation contours. Next, we align the resulting contours for each class with respect to similarity transformations (scale, rotation and translation) using a genetic algorithm [51], whose numeric implementation is by courtesy of Dr. Xavier Bresson.

Afterwards, we use the aligned contours corresponding to each class to estimate the parameters of our probability models. Namely, for the Gaussian probabilities, we employ the method described in [126] to obtain smooth estimates of the mean ϕ_i and variance σ_i for each class S_i. That is, we wish to maximize the joint probability of the training samples $\phi_1, \phi_2, \ldots, \phi_K$ with respect to the model parameters ϕ_i and variance σ_i:

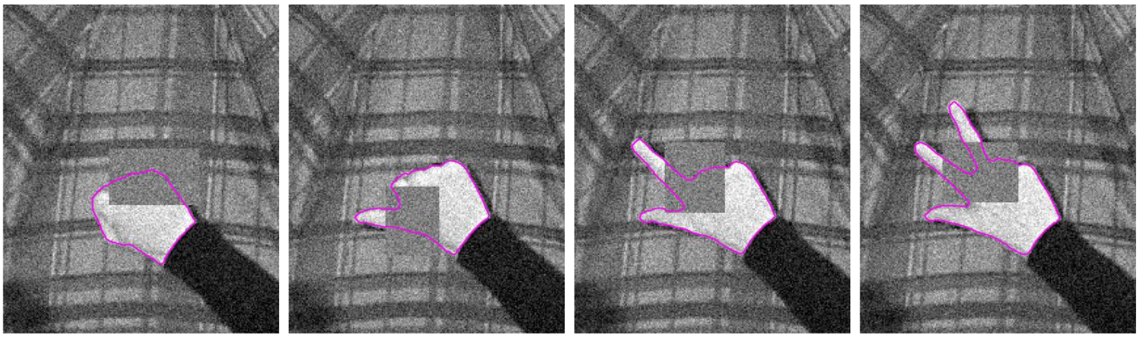
$$\begin{aligned} (\phi_i, \sigma_i) &= \arg\max_{\phi,\sigma} P(\phi_1, \ldots, \phi_K|\phi, \sigma) \\ &= \arg\max_{\phi,\sigma} \prod_{k=1}^{K} P(\phi_k|\phi, \sigma), \end{aligned} \tag{4.7}$$



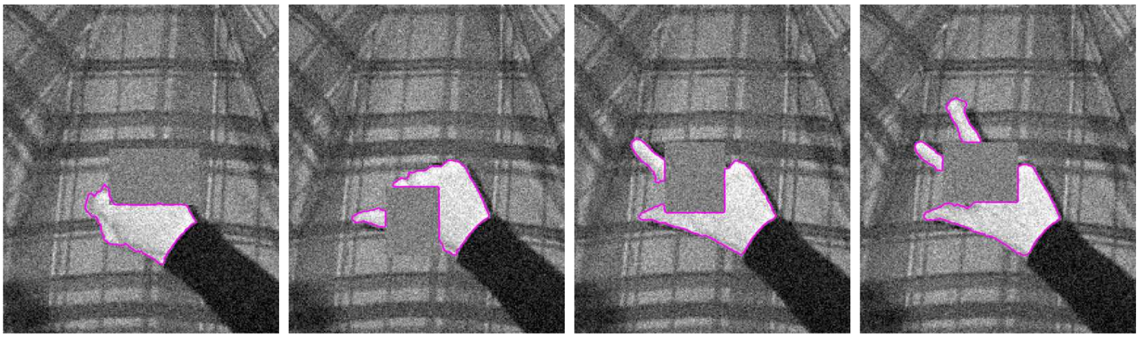
(a) Seq. 1 Fr. 2 (b) Seq. 1 Fr. 26 (c) Seq. 1 Fr. 51 (d) Seq. 1 Fr. 80



(e) Seq. 1 Fr. 2 (f) Seq. 1 Fr. 26 (g) Seq. 1 Fr. 51 (h) Seq. 1 Fr. 80



(i) Seq. 2 Fr. 2 (j) Seq. 2 Fr. 22 (k) Seq. 2 Fr. 68 (l) Seq. 2 Fr. 100



(m) Seq. 2 Fr. 2 (n) Seq. 2 Fr. 22 (o) Seq. 2 Fr. 68 (p) Seq. 2 Fr. 100

Figure 4.2 — (a)–(d), (i)–(l) Segmentation with the proposed implementation of our general framework (using Gaussian probability models) of two image sequences in the presence of occlusion, background complexity and noise (second sequence). (e)–(h), (m)–(p) Conventional segmentation of the same image sequences. For the latter, we used the Chan-Vese piecewise-constant segmentation model (the image-based term of our energy (4.2)).

where in the last line we assume independence of the training samples. Applying the negative logarithm and substituting $P(\phi_k|\phi,\sigma)$ with (4.4), (4.7) becomes:

$$(\phi_i,\sigma_i) = \arg\min_{\phi,\sigma} -\sum_{k=1}^{K} \iint_\Omega \log p^{(x,y)}(\phi_k|\phi,\sigma)\,dx\,dy, \tag{4.8}$$

which is the formulation of a functional which can be minimized through variational methods to obtain ϕ_i and σ_i. In order to impose spatial coherence over the variance estimates throughout the image, we add a smoothness constraint to the energy, yielding:

$$\begin{aligned} E(\phi,\sigma) = &\sum_{k=1}^{K} \iint_\Omega \log\sigma(x,y) + \frac{(\phi_k(x,y)-\phi(x,y))^2}{2\sigma^2(x,y)}\,dx\,dy \\ &+ \gamma \iint_\Omega |\nabla\sigma(x,y)|^2\,dx\,dy. \end{aligned} \tag{4.9}$$

This results in the following evolution equations for the mean ϕ and variance σ:

$$\begin{aligned} \frac{\partial\phi}{\partial t}(x,y) = &\sum_{k=1}^{K} \frac{\phi_k(x,y)-\phi(x,y)}{2\sigma^2(x,y)}, \\ \frac{\partial\sigma}{\partial t}(x,y) = &\sum_{k=1}^{K} -\frac{1}{\sigma(x,y)} + \frac{(\phi_k(x,y)-\phi(x,y))^2}{\sigma^3(x,y)} + 2\gamma\triangle\sigma(x,y). \end{aligned} \tag{4.10}$$

Since we wish to obtain the mean level set function ϕ as a signed distance function, we insert a step of re-initialization of ϕ to the signed distance function after each evolution step by (4.10), following the method of [1]. To obtain our final estimates of the mean ϕ_i and variance σ_i for each class S_i, we run equations (4.10), interleaved with the re-initialization step, until we reach their steady state.

In order to assess the action class initial and transition probabilities (3.3) and (3.5), we estimate the relative occurrence frequency of starting classes and of transitions between classes from the training sequences. Using Bayes' rule, the transition probability from a class S_i to a class S_j can be written as:

$$t_{ij} = P(s_t = S_j|s_{t-1}=S_i) = \frac{P(s_t=S_j, s_{t-1}=S_i)}{P(s_{t-1}=S_i)}. \tag{4.11}$$

Considering a set of N training sequences of labeled gestures, each of length T_n, $n = 1..N$, we estimate the probabilities from the right-hand side of (4.11) as:

$$\begin{aligned} P(s_{t-1}=S_i) &= \frac{\sum_{n=1}^{N}\sum_{t=2}^{T_n} \delta(s_{t-1}^n, S_i)}{\sum_{n=1}^{N}\sum_{t=2}^{T_n} 1}, \\ P(s_t=S_j, s_{t-1}=S_i) &= \frac{\sum_{n=1}^{N}\sum_{t=2}^{T_n} \delta(s_{t-1}^n, S_i)\,\delta(s_t^n, S_j)}{\sum_{n=1}^{N}\sum_{t=2}^{T_n} 1}. \end{aligned} \tag{4.12}$$

Here, s_t^n is the class label of frame t from the n-th training sequence and δ is the Kronecker-delta symbol: $\delta(s, S) = \begin{cases} 1 & \text{if } s = S, \\ 0 & \text{otherwise} \end{cases}$. Thus, the transition probability is given by:

$$t_{ij} = \frac{\sum_{n=1}^{N} \sum_{t=2}^{T_n} \delta(s_{t-1}^n, S_i)\, \delta(s_t^n, S_j)}{\sum_{n=1}^{N} \sum_{t=2}^{T_n} \delta(s_{t-1}^n, S_i)}. \tag{4.13}$$

Similarly, the initial action class probabilities are given by:

$$\pi_i = \frac{\sum_{n=1}^{N} \delta(s_1^n, S_i)}{\sum_{n=1}^{N} 1} = \frac{\sum_{n=1}^{N} \delta(s_1^n, S_i)}{N}. \tag{4.14}$$

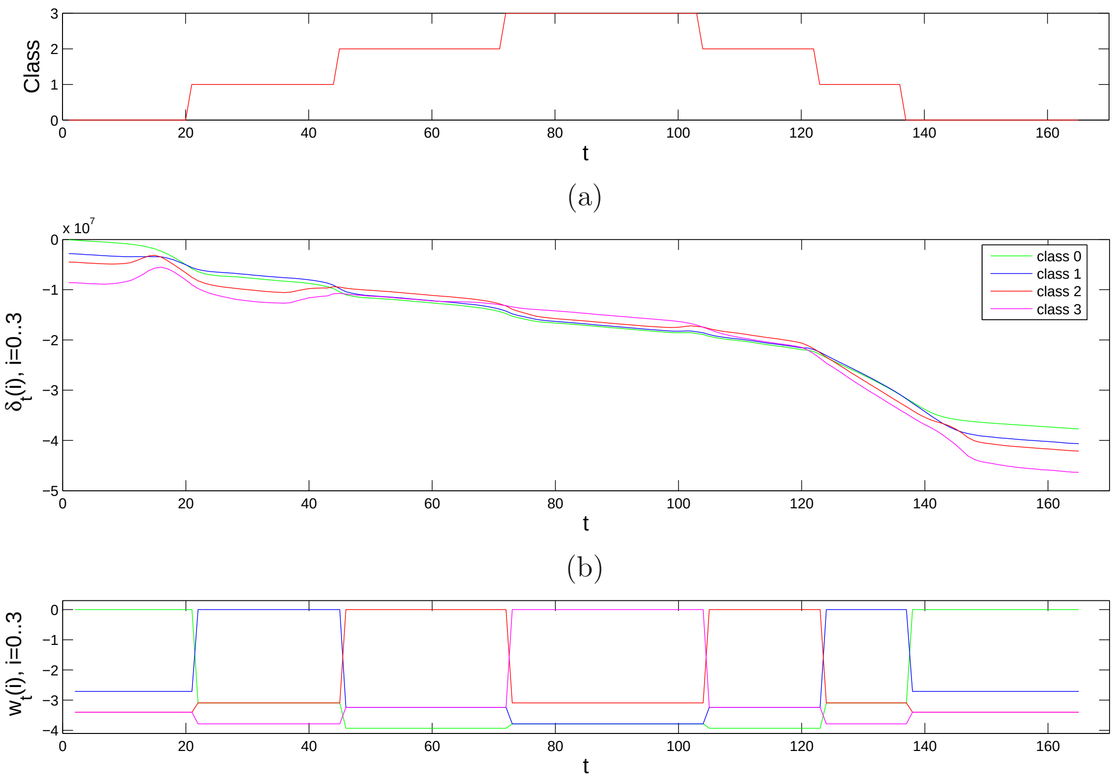
Alternatively, one can estimate these parameters (the class mean and variance, as well as the action class initial and transition probabilities) using an expectation-maximization (EM) approach. More precisely, since once the attributes are observed, our model consists of an HMM, the parameter estimation can be performed via the Baum-Welch algorithm (see [11, 120]).

4.2.4 Experimental Results

We tested the implementation of our framework, resulting after model instantiation and training, on new gesture image sequences of a counting hand. In particular, we used the succession of gestures 0,1,2,3,2,1,0, performed in front of a complex background and degraded by occlusions. The segmentation contour for the first image of each sequence has been determined by a manual initialization in the proximity of the hand, followed by segmentation using only the image-based term of our segmentation energy (4.2). The parameters for the variational segmentation were $\alpha = 5000$ and $\nu = 4000$. The average execution time using un-optimized code (Matlab and C) was 3-4 minutes per frame.

Our framework brings considerable improvements to the segmentation/behavior recognition task, even by modeling class contour characteristics via the unsophisticated Gaussian probability model. By virtue of the prior information supplied by the recognition process, segmentation is able to cope with severe occlusions, as can be seen in Fig. 4.2 (a)–(d), (i)–(l). For comparison, Fig. 4.2 (e)–(h), (m)–(p) shows the results obtained on the same sequence with conventional segmentation, i.e., by using the Chan-Vese piecewise-constant model (the image-based term of our model (4.2)). The latter are clearly inferior, since the desired shape of the object cannot be recovered because of the occlusions.

Figure 4.3 shows the recognition results for the first test sequence, which correctly follow the test gesture sequence and our understanding of it in terms of the executed gestures. Moreover, the frame classification obtained by backtracking from the recognition process corresponds to the partial classification results obtained throughout the sequence, which have been used to guide segmentation. This concordance can be seen in Fig. 4.3, which exhibits, as functions of time (frame), (a) the final classification, (b) the logarithm of the $\tilde{\delta}$ quantities for each gesture class, and (c) the log prior confidence of each class (the logarithm of $w_t(S_i)$) used as input to the segmentation. Thus, online recognition (yielding



(a)

(b)

(c)

Figure 4.3 — Behavior recognition results plotted per frame. (a) Final frame classification. (b) Logarithm of the $\tilde{\delta}$ quantities for each class. (c) Logarithm of $w_t(S_i)$ — the prior confidence of each class S_i, used as input to the segmentation. The logarithm values are scaled with respect to their maximum value for each frame.

classification results at each frame) is possible within our framework and also yields the correct recognition results.

4.3 A Finger-spelling Application

4.3.1 Application Description

The second application that we used to test our framework focuses on finger-spelling recognition. As explained in the introductory section of this chapter, finger-spelling is a component of sign language which consists of manual representations of alphabet letters. Therefore, the gesture classes involved in our application correspond to these manual letter descriptions. This makes it more challenging than our finger-counting application, since it entails a larger number of classes and poorer discrimination among them, as we will see in the following.

We use the manual alphabet of the French-speaking part of Switzerland (Suisse Romande) [63], which is depicted in Fig. 4.4. As can be seen in Fig. 4.4, the gestures corresponding to different letters are not easy to differentiate, with letter pairs such as (A, S), (M, N) or (R, U) easily confoundable. In this context, our goal is to perform finger-spelling recognition on a 15-word vocabulary containing country names, as presented in Table 4.1.

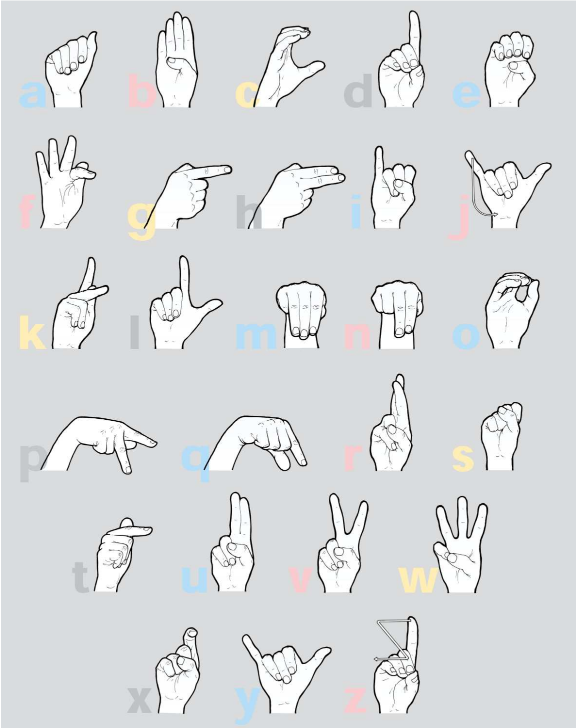


Figure 4.4 — Manual alphabet of the French-speaking part of Switzerland. Reproduced from [63].

With the support of the Swiss Federation for the Hearing-Impaired (Fédération Suisse des Sourds) [63], we have acquired a data base containing image sequences of a hearing-impaired person finger-spelling the above mentioned words. Acquisition has been performed both in ideal conditions (contrasting background, low speed gesturing), for training purposes, and realistic ones (cluttered background, normal speed gesturing), for testing purposes. The two acquisition scenarios are illustrated in Fig 4.5, (a) and (b) respectively.

Table 4.1 — Vocabulary of our finger-spelling application

ALBANIA	ALGERIA	ARMENIA	AUSTRIA	BELARUS
BELGIUM	BURUNDI	CROATIA	DENMARK	ECUADOR
ERITREA	ESTONIA	FINLAND	GEORGIA	GERMANY

4.3.2 Solution Based on the Proposed Framework

For this application, we use the same object attribute as for our finger-counting application, i.e., the hand contour, represented via the level set function ϕ: $f_A(I_t, C_t) = \phi(I_t, C_t)$, where $\phi : \Omega \to \mathbb{R}$. As before, ϕ is given by the signed distance function to the hand contour C_t,

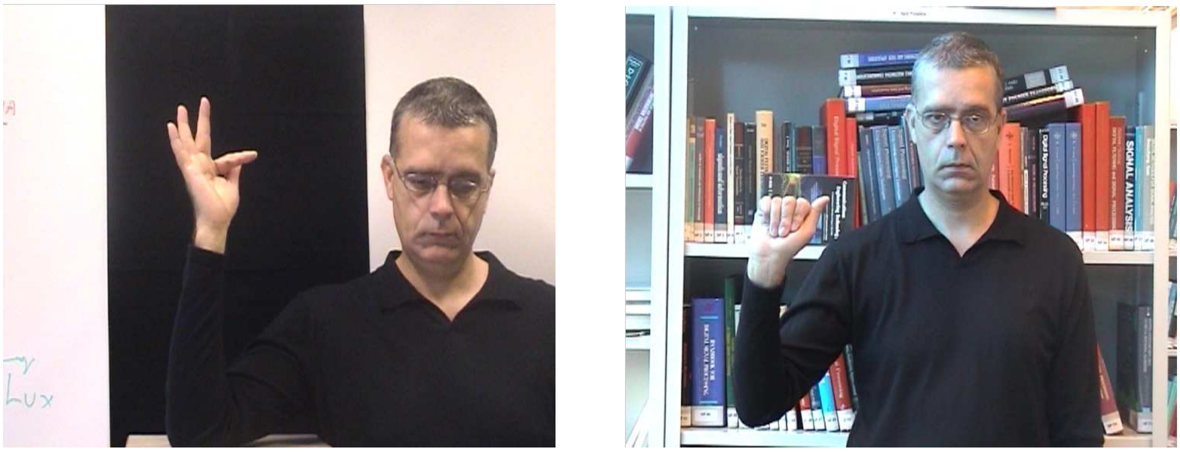

(a) (b)

Figure 4.5 — Image sequence acquisition for our finger-spelling application, with the aid of a hearing-impaired person from the Swiss Federation for the Hearing-Impaired. (a) Acquisition for model training, on contrasting background. (b) Acquisition for testing, on cluttered background.

according to (4.1). Moreover, the probabilistic and segmentation models corresponding to time slice t will be expressed in terms of the function ϕ, which encapsulates the contour C_t.

In order to overcome the difficulty in discriminating among the different letter classes in our alphabet (as seen in Fig. 4.4), we make use of the supplementary knowledge regarding the allowed words, that is, the ones belonging to the vocabulary in Table 4.1). To this end, we employ the extension of our framework designed for the recognition of a predefined behavior set (presented in Section 3.9 of Chapter 4), enabling us to introduce constraints regarding the allowed behavior types.

For the case of our application, the behavior types correspond to the 15 words in our vocabulary. Each of the words can be decomposed into its basic components — the letters — which are shared among all words and constitute the action/gesture classes of our model. Thus, we have a total of 18 action classes, corresponding to the 18 letters making up the chosen vocabulary: A, B, C, D, E, F, G, I, K, L, M, N, O, R, S, T, U, Y.

Regarding the class probability models, a limitation of the Gaussian model that we used in our first application is the fact that the mean and variance of the prior corresponding to each class are fixed throughout the image sequence, and thus cannot adapt to varying shapes of the same class. This makes it difficult to obtain accurate segmentations for images where the winning class prior doesn't offer a close match to the image, even after the similarity transformation. For this application, we obtain an improvement with respect to this limitation by using PCA-based probability models, which adapt dynamically to the content of new images, as we describe in the following.

The probability model $P_i(\phi)$ corresponding to class S_i relies on a shape distance function between the segmenting contour and a prior contour corresponding to that class, motivated by [23]. The prior contours for each class are computed via principal components analysis (PCA) from specific training data for each class. They evolve during segmentation so as best to match image information, within class constraints imposed by the PCA. We improve the distance function proposed in [23] by making it symmetric, resulting into probability models which are suitable for classification. Symmetry in the construction of shape priors for level set functions is advocated in [41].

In the context of level-set-based variational image segmentation, the PCA representation is promoted by approaches such as [23, 40, 89]. The purpose of PCA is to reduce redundant information and summarize the main variations of a training set. It is mathematically defined as an orthogonal linear transform, that transforms the data to a new coordinate system, where the greatest variance is obtained by projecting the data onto the first coordinate, the second greatest variance — by projecting it on the second coordinate, and so on. In this way, the dimensionality of the data can be reduced by retaining only those data characteristics which mostly contribute to its variance.

More precisely, given a training set of level set functions, which have been discretized on a rectangular grid and arranged in vector format $\{\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n\}, \quad \boldsymbol{\phi}_k \in \mathbb{R}^m$, its principal directions of variation are captured by the eigenvectors $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}, \quad \mathbf{e}_k \in \mathbb{R}^m$ of the covariance matrix $\boldsymbol{\Sigma} = \frac{1}{n-1}\mathbf{M}\,\mathbf{M}^\top$. The column vectors of the matrix $\mathbf{M}$ are the n mean-centered training level set functions, obtained by extracting the mean $\overline{\boldsymbol{\phi}} = \frac{1}{n}\sum_{k=1}^{n} \phi_k$ from each training sample $\boldsymbol{\phi}_k$. Given the eigen-decomposition of the covariance matrix $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, an approximate representation $\mathbf{M}^*$ of the training data $\mathbf{M}$ can then be obtained in the reduced space of the $p < m$ eigenvectors $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$, which are the columns of $\mathbf{U}$ corresponding to the p largest eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}$: $\mathbf{M}^* = \mathbf{E}^\top\mathbf{M}$, with $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_p]$. In particular, each training sample ϕ_k is reduced to $\mathbf{c}_k = \mathbf{E}^\top(\phi_k - \overline{\boldsymbol{\phi}})$, thus $\mathbf{M}^* = [\mathbf{c}_1, \dots, \mathbf{c}_n]$. A new level set function $\hat{\boldsymbol{\phi}}$ can then be approximated with respect to the extracted PCA eigenvectors as:

$$\hat{\boldsymbol{\phi}} = \overline{\boldsymbol{\phi}} + \mathbf{E}\,\mathbf{c}. \tag{4.15}$$

Here $\mathbf{c}$ is the p-dimensional vector of eigen-coefficients, which constitutes the reduced representation of $\hat{\boldsymbol{\phi}}$.

Therefore, a PCA-prior contour, corresponding to a particular gesture class, can be represented in terms of the PCA coefficients $\mathbf{c}$, using the class-specific mean level set function and eigenvectors. These enable us to obtain the level set function of the prior contour $\hat{\phi}$, as the continuous interpolation throughout the image domain of the discrete level set function $\hat{\boldsymbol{\phi}}$, computed from the PCA coefficients $\mathbf{c}$ as in (4.15). Moreover, similarly to our finger-counting application, we introduce the alignment of a prior contour with respect to the current segmenting contour. This alignment is in terms of similarity transformations acting on the image domain $h_{\boldsymbol{\tau}}(x, y)$, as in (4.5). Such a transformation is parameterized by the vector $\boldsymbol{\tau} = \{s, \theta, T_x, T_y\}$, where s represents scale, θ is the rotation angle and T_x, T_y are the x- and y-axis translations, respectively. Thus, we obtain the level set function of the prior contour $\hat{\phi}(\mathbf{c}, \boldsymbol{\tau})$ from its class-specific PCA and alignment parameters $\mathbf{c}$ and $\boldsymbol{\tau}$, as the interpolation of

$$\hat{\boldsymbol{\phi}}(\mathbf{c}, \boldsymbol{\tau}) = \frac{1}{s}\left(\overline{\boldsymbol{\phi}}(h_{\boldsymbol{\tau}}(x, y) + \mathbf{E}(h_{\boldsymbol{\tau}}(x, y))\,\mathbf{c}\right). \tag{4.16}$$

In this context, we define our shape distance function between the current segmenting contour ϕ and the prior contour $\hat{\phi}$, with the latter being parameterized by $\mathbf{c}$ and $\boldsymbol{\tau}$, as

$$d(\phi, \mathbf{c}, \boldsymbol{\tau}) = \iint_\Omega \left(\hat{\phi}^2(\mathbf{c}, \boldsymbol{\tau}) \, |\nabla \phi| \, \delta(\phi) + \phi^2 \, |\nabla \hat{\phi}(\mathbf{c}, \boldsymbol{\tau})| \, \delta(\hat{\phi}(\mathbf{c}, \boldsymbol{\tau})) \right) dx \, dy, \tag{4.17}$$

where δ is the Dirac function. Since $\iint_\Omega |\nabla \phi| \, \delta(\phi) \, dx \, dy$ represents the length of the zero level set of ϕ and the level set functions are represented as signed distance functions, we readily observe that the first term of (4.17) approximates the minimal Euclidian distance to the prior contour, integrated along the segmenting contour. This is an approximation because the level set function $\hat{\phi}$ resulting from PCA is not the exact distance function, but just a reasonable approximation of it. The second term of (4.17) exchanges the roles of ϕ and $\hat{\phi}$ relative to the first term, making the distance function symmetric and thus suitable for use in classification. Based on this distance function, we define the probability of the segmenting contour represented by ϕ, corresponding to class S_i, as

$$P_i(\phi) \propto e^{-d(\phi, \mathbf{c}^i, \boldsymbol{\tau}^i)}. \tag{4.18}$$

As image- and contour-dependent terms, guiding the evolution of the main contour ϕ and prior contours $\hat{\phi}_i(\mathbf{c}^i, \boldsymbol{\tau}^i)$ (in terms of their parameters $\mathbf{c}^i$ and $\boldsymbol{\tau}^i$), we use the piecewise constant Chan-Vese model [30], adapted to color images given by the red, green and blue components $I(x, y) = (I^R(x, y), I^G(x, y), I^B(x, y))$:

$$\begin{aligned} & E_{\text{image}}(I_t, \phi) + E_{\text{image}}(I_t, \mathbf{c}^{i=1..M}, \boldsymbol{\tau}^{i=1..M}) + \nu E_{\text{contour}}(\phi) \\ & \qquad = \sum_{k \in \{R,G,B\}} \lambda_k \iint_\Omega (I_t^k - \mu_{\phi+}^k)^2 H(\phi) + (I_t^k - \mu_{\phi-}^k)^2 H(-\phi) \, dx \, dy \\ & \qquad + \sum_{k \in \{R,G,B\}} \lambda_k \sum_{i=1}^{M} \iint_\Omega (I_t^k - \mu_{\hat{\phi}_i+}^k)^2 H(\hat{\phi}_i) + (I_t^k - \mu_{\hat{\phi}_i-}^k)^2 H(-\hat{\phi}_i) \, dx \, dy \\ & \qquad + \nu \iint_\Omega |\nabla H(\phi)| \, dx \, dy. \end{aligned} \tag{4.19}$$

Here H is the Heaviside function, $\mu_{\phi+}^k$, $\mu_{\hat{\phi}_i+}^k$ and $\mu_{\phi-}^k$, $\mu_{\hat{\phi}_i-}^k$ are the mean values of the k-th component of the image vector ($k \in \{R, G, B\}$) over the positive, respectively negative, regions of the level set functions ϕ and $\hat{\phi}_i$. The ratio between the RGB components is given by the weights $\lambda_k \geq 0$, $k \in \{R, G, B\}$. Function $\hat{\phi}_i = \hat{\phi}_i(\mathbf{c}^i, \boldsymbol{\tau}^i)$ is the continuously interpolated level set function of the prior contour (4.16), and the last term of (4.19) imposes smoothness of contour ϕ.

The prior term of the energy is obtained from (3.37) by using the prior information corresponding to our modified model $P(A_t|s_t) \, \tilde{w}_t(s_t)$, as described in Section 3.9 of Chapter 3. The probabilities $P(A_t|s_t = S_i)$ are substituted with $P_i(\phi)$ (4.18), yielding:

$$E_{\text{prior}}(\phi, \mathcal{L}, \mathbf{c}^{i=1..M}, \boldsymbol{\tau}^{i=1..M}) = \sum_{i=1}^{M} \left(-\log \tilde{w}_t(S_i) + d(\phi, \mathbf{c}^i, \boldsymbol{\tau}^i) \right) L_i^2 + \beta \left(1 - \sum_{i=1}^{M} L_i^2 \right)^2. \tag{4.20}$$

Towards computational efficiency, we adopt a pruning strategy, using only the top 4 most probable priors (out of the 18 available priors) to guide the segmentation of each image. These top 4 prior letters are chosen using the maximum prior letter probabilities, computed with (3.88). Our pruning strategy does not affect recognition performance, while diminishing segmentation time and improving convergence towards the optimal prior. This constitutes an advantage of our dynamic framework over the simple use of competing multiple priors for the segmentation of each image, which would imply the simultaneous optimization and competition between 18 priors, with little chances of convergence towards the optimum prior due to local minima.

The total energy (3.35), summing (4.19) and (4.20), is minimized via the calculus of variations and gradient descent. The evolution equations for the level set function ϕ, the labels $\mathcal{L}$, the PCA and alignment parameters $\mathbf{c}^i$ and $\boldsymbol{\tau}^i$, are presented in Section A.3 from the Appendix of this thesis.

4.3.3 Management of Co-articulation Effects

A frame-by-frame inspection of sample image sequences in our finger-spelling application reveals the fact that a considerable part of these are actually frames of transition between the spelling of different letters. The hand configurations in these frames do not correspond to the gesture class models for any of the letters, or, even worse, they may match the models of spurious letters that are not actually present in the finger-spelt word. This kind of gesture modifications under the influence of neighboring gestures is similar to the phenomenon of co-articulation in phonetics.

Obviously, these co-articulation effects are an impediment to the gesture recognition task, potentially causing erroneous classification. A solution used in the field of speech recognition is to include the co-articulated part in the modeling, by creating models of pairs or triplets of sounds. In our framework, a possibility would be to model the frames of transition between various letter pairs as different classes. However, such a solution would be prohibitive in terms of computation and would not scale up with a large number of letter / gesture classes (due the large number of possible letter pairs). Therefore, we adopted a more pragmatic solution, which involves a slight modification of our framework.

The main idea of our modification is to discard the transition frames from the behavior estimation and rely solely on the frames belonging the gestures themselves. To this end, we implement a mechanism for detecting transition frames, based on the monitoring of the momentary best estimate for the letter class of a certain frame. In our practical experiments, we noticed that this estimate remains unchanged throughout the main gesture duration, and changes (as we would expect) at the moment of the gesture transition. Birk et al. [13] also aim at discarding transition frames from the recognition task, but adopt a different technique for detecting them, based on monitoring the amount of motion between frames.

In our implementation, after detecting a gesture transition, we enter into a transition phase, with a limited maximum duration (given as a number of frames). During this phase, we do not compute the values of our variables $\tilde{\delta}_t(s_t, b)$ and $\psi_t(s_t, b)$ recursively, in the usual manner, but by relying on their last valid estimates, obtained before the transition period. The determined $\tilde{\delta}$ and ψ values for each transition frame are used to guide the segmentation of the respective frames, but are discarded from the final behavior estimation by backtracking. Therefore, in the final classification, the transition frames are assigned the label "transition". Their adjacent frames are classified by referring to the closest validly classified frame.

To exit the transition phase before its maximum duration, we search for evidence of stable letter recognition. That is, if the estimation of the best momentary letter class remains constant for a given number of consecutive frames, we consider that the gesture has become stable and we return to the normal course of computation. To this end, we look for the first occurrence of the stable letter class and recompute the past values of the $\tilde{\delta}$ and ψ variables, up to the current frame where we exited the transition phase. We also remove the transition status of the frames corresponding to the stability period, so that their $\tilde{\delta}$ and ψ values can be used in the final classification by backtracking. In the case where we did not detect stable letter recognition, we return to normal computation after the given maximum transition duration. To compute the new $\tilde{\delta}$ and ψ values, we rely on their last valid estimates before the transition. To eliminate some cases of spurious transition detections, we also exit the transition phase if we detect a transition to the last valid letter estimated before the transition (self-transition). In this case, we compute all the past $\tilde{\delta}$ and ψ values up to the current frame, and remove the transition status of the respective frames.

To summarize, we present a schematic description of our modified algorithm. To this end, we introduce the constants $MaxTransFrames$ to denote the maximum duration of the transition period, and *StabilityThreshold* to denote the number of consecutive frames where the momentary gesture estimation needs to be stable, in order to exit the transition period. Additionally, we introduce the variable *transitionStep*(t) to indicate whether frame t is a transition frame and if so, to store the transition step count: $1 \leq$ *transitionStep*$(t) \leq$ $MaxTransFrames$ if t is a transition frame and *transitionStep*$(t) = 0$ otherwise. Moreover, we use the variable *winnerClass*(t) to store the best momentary estimate for the letter class corresponding to frame t. Denoting by $NPriors$ the reduced number of class priors used to guide image segmentation after pruning, we use the set *topS*$(t) \subset S$ to hold the $NPriors$ top most probable letter classes s_t at time t. The joint segmentation and recognition of a new test sequence $I_{1:T}$ proceeds as follows [*].

1. Extract contour $\tilde{C}_1^*$ by segmenting image I_1, according to the considerations given in Chapter 3.
2. Estimate $\tilde{\delta}_1(s_1, b)$, $s_1 \in S$, $b \in B$, according to (3.91).
3. Compute best momentary estimate for the letter class *winnerClass*$(1) = \arg\max_{s_1} \max_b \tilde{\delta}_1(s_1, b)$.

---

[*] For notation correspondence with Chapter 3, we write probabilities and variables in terms of C_t, and imply the use of their corresponding expressions in terms of the level set function ϕ, given in the current chapter.

```
4. Compute $w_2(s_2, b)$ and $\tilde{w}_2(s_2)$ for $s_2 \in S$, $b \in B$, using (3.86) and (3.88), respectively.
5. Compute top $NPriors$ most probable priors for next frame:
   $topS(2) = \{s_2 \in S,\ s_2 \in \text{top } NPriors \text{ ranked by } \tilde{w}_2(s_2)\}$.
6. Set $I_1$ as a non-transition frame: $transitionStep(1) = 0$.
7. for t = 2 to T
    (a) Estimate contour $\tilde{C}_t^*$ by segmenting image $I_t$ using energy (3.35), composed of
        (4.19) and (4.20), where $E_{\text{prior}}$ includes the priors of the classes $s_t \in topS(t)$.
        Initialize segmentation with $\tilde{C}_{t-1}^*$.
    (b) Compute current $\tilde{\delta}_t(s_t, b)$ and $winnerClass(t)$ and update current transition
        state $transitionStep(t)$.
    (c) Compute $w_{t+1}(s_{t+1}, b)$, $\psi_{t+1}(s_{t+1}, b)$, $\tilde{w}_{t+1}(s_{t+1})$ and $topS(t+1)$.
   end
8. Estimate optimal behavior type and backtrack to infer the optimal letter class sequence $s_{1..T}^*$.
```

Step 7(b) of our algorithm can be detailed as follows:

```
$\tilde{\delta}_t(s_t, b) = P(I_t, \tilde{C}_t^* | s_t)\, w_t(s_t, b),\ s_t \in S,\ b \in B$;
$winnerClass(t) = \arg\max_{s_t} \max_b \tilde{\delta}_t(s_t, b)$;
if $0 < transitionStep(t-1) < MaxTransFrames$
  • Get time of last frame before the transition:
    $tValid$ = largest $t$ with $transitionStep(t) == 0$;
  • if $winnerClass(t) \neq winnerClass(tValid) \Rightarrow$ true transition:
      – $transitionStep(t) = transitionStep(t-1) + 1$;
      – if $winnerClass(t) == winnerClass(t-1) \Rightarrow$ stable letter:
          ∗ $stabilityCounter = stabilityCounter + 1$;
      – else
          ∗ $stabilityCounter = 1$;
      – end
      – if $stabilityCounter == StabilityThreshold \Rightarrow$ reached letter stability threshold,
        exit transition phase and compute past $\tilde{\delta}$-s for the stability period:
          ∗ $transitionStep(t) = 0$;
          ∗ Get beginning time of stability period: $tStable = t - StabilityThreshold + 1$;
          ∗ while $tStable < t$
```

```
            · for $s_{tStable} \in S,\ b \in B$
                $w_{tStable}(s_{tStable}, b) = \max_{s_{tValid}} \tilde{\delta}_{tValid}(s_{tValid}, b)\, P(s_{tStable}|s_{tValid}, b);$
                $\psi_{tStable}(s_{tStable}, b) = \arg\max_{s_{tValid}} \tilde{\delta}_{tValid}(s_{tValid}, b)\, P(s_{tStable}|s_{tValid}, b);$
                $\tilde{\delta}_{tStable}(s_{tStable}, b) = P(I_{tStable}, \tilde{C}^{*}_{tStable}|s_{tStable})\, w_{tStable}(s_{tStable}, b);$
            · end
            · $winnerClass(tStable) = \arg\max_{s_{tStable}} \max_{b} \tilde{\delta}_{tStable}(s_{tStable}, b);$
            · $transitionStep(tStable) = 0;$
            · $tValid = tStable;$
            · $tStable = tStable + 1;$
        * end
        * Compute $w$ and $\psi$ values for current frame:
          $w_t(s_t, b) = \max_{s_{t-1}} \tilde{\delta}_{t-1}(s_{t-1}, b)\, P(s_t|s_{t-1}, b),\ s_t \in S,\ b \in B;$
          $\psi_t(s_t, b) = \arg\max_{s_{t-1}} \tilde{\delta}_{t-1}(s_{t-1}, b)\, P(s_t|s_{t-1}, b),\ s_t \in S,\ b \in B;$
    – end
• else ⇒ self-transition, cancel transition phase, recompute past $\tilde{\delta}$-s:
    – $transitionStep(t) = 0;$
    – $tTrans = tValid + 1;$
    – while $tTrans < t$
        * for $s_{tTrans} \in S,\ b \in B$
            $w_{tTrans}(s_{tTrans}, b) = \max_{s_{tTrans-1}} \tilde{\delta}_{tTrans-1}(s_{tTrans-1}, b)\, P(s_{tTrans}|s_{tTrans-1}, b);$
            $\psi_{tTrans}(s_{tTrans}, b) = \arg\max_{s_{tTrans-1}} \tilde{\delta}_{tTrans-1}(s_{tTrans-1}, b)\, P(s_{tTrans}|s_{tTrans-1}, b);$
            $\tilde{\delta}_{tTrans}(s_{tTrans}, b) = P(I_{tTrans}, \tilde{C}^{*}_{tTrans}|s_{tTrans})\, w_{tTrans}(s_{tTrans}, b);$
        * end
        * $winnerClass(tTrans) = \arg\max_{s_{tTrans}} \max_{b} \tilde{\delta}_{tTrans}(s_{tTrans}, b);$
        * $transitionStep(tTrans) = 0;$
        * $tTrans = tTrans + 1;$
    – end
    – Compute $w$ and $\psi$ values for current frame:
      $w_t(s_t, b) = \max_{s_{t-1}} \tilde{\delta}_{t-1}(s_{t-1}, b)\, P(s_t|s_{t-1}, b),\ s_t \in S,\ b \in B;$
      $\psi_t(s_t, b) = \arg\max_{s_{t-1}} \tilde{\delta}_{t-1}(s_{t-1}, b)\, P(s_t|s_{t-1}, b),\ s_t \in S,\ b \in B;$
• end
else ⇒ not in transition phase, detect possible transition:
    • if $winnerClass(t) \neq winnerClass(t-1) \Rightarrow$ transition detected:
        – $transitionStep(t) = 1;$
        – $stabilityCounter = 1;$
    • else
        – $transitionStep(t) = 0;$
    • end
end
```

Step 7(c) of our algorithm consists of the following:

```
if $0 < transitionStep(t) \leq MaxTransFrames \Rightarrow$ transition phase, transmit past information to next stage:
    • $w_{t+1}(s,b) = w_t(s,b),\ s \in S,\ b \in B;$
    • $\psi_{t+1}(s,b) = \psi_t(s,b),\ s \in S,\ b \in B;$
    • $\tilde{w}_{t+1}(s) = \tilde{w}_t(s),\ s \in S;$
    • $topS(t+1) = topS(t);$
else $\Rightarrow$ prepare $w$ and $topS$ for next frame:
    • $w_{t+1}(s_{t+1},b) = \max_{s_t}\left(\tilde{\delta}_t(s_t,b)\,P(s_{t+1}|s_t,b)\right),\ s_{t+1} \in S,\ b \in B;$
    • $\psi_{t+1}(s_{t+1},b) = \arg\max_{s_t}\left(\tilde{\delta}_t(s_t,b)\,P(s_{t+1}|s_t,b)\right),\ s_{t+1} \in S,\ b \in B;$
    • $\tilde{w}_{t+1}(s_{t+1}) = \max_b w_{t+1}(s_{t+1},b);$
    • $topS(t+1) = \{s_{t+1} \in S,\ s_{t+1} \in$ top $NPriors$ ranked by $\tilde{w}_{t+1}(s_{t+1})\}.$
end
```

Step 8 of our algorithm is performed as follows:

```
if $transitionStep(T) == 0$
    • $tValid = T;$
else
    • Get time of last frame which did not belong to a transition phase:
      $tValid =$ largest $t$ with $transitionStep(t) == 0;$
end
Determine winner behavior type: $b^* = \arg\max_b \max_{s_{tValid}} \tilde{\delta}_{tValid}(s_{tValid},b);$
$tValidLast = tValid;$
for $t = tValid - 1$ down to $1$
```

```
• if $transitionStep(t) \neq 0$
    – $s_t^* = transition$;
• else
    – if $transitionStep(t+1) == 0$
        * $s_t^* = \psi_{t+1}(s_{t+1}^*, b^*)$;
    – else
        * $s_t^* = \psi_{tValidLast}(s_{tValidLast}^*, b^*)$;
    – end
    – $tValidLast = t$;
• end
end
```

4.3.4 Database and Training of the Model

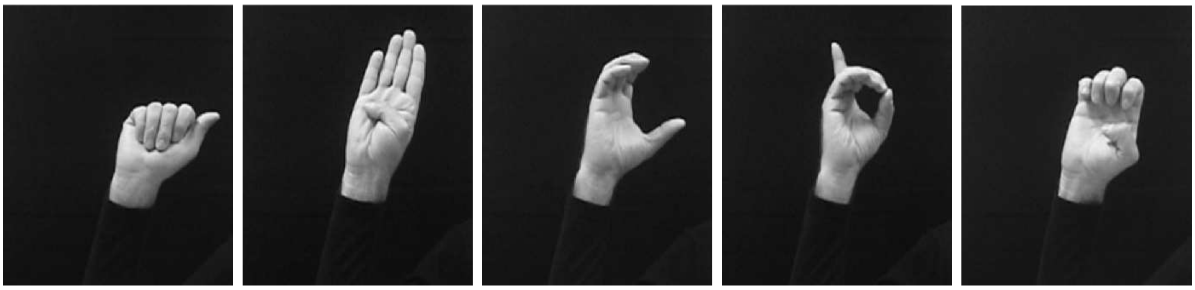
We trained our model using image sequences of each vocabulary word from the acquired database. For training, the gesturing person was filmed on a dark, contrasting background and the gestures were performed at slow speed. Figure 4.6 presents images from the training sequences.

First, the gesturing hand was segmented in each training sequence and the resulting contours were assigned to their respective letter classes and aligned with respect to similarity transformations (scale, rotation and translation) using genetic algorithms [51]. Subsequently, the parameters of the observation probability model $P(A_t|s_t = S_i) = P_i(\phi)$ (4.18) for each letter class S_i were learned by PCA ($p = 20$) separately from the training contours of the respective letter class. This resulted in a corresponding mean $\overline{\boldsymbol{\phi}}_i$ and eigenvectors $\mathbf{E}_i$ for each letter/behavior class S_i.

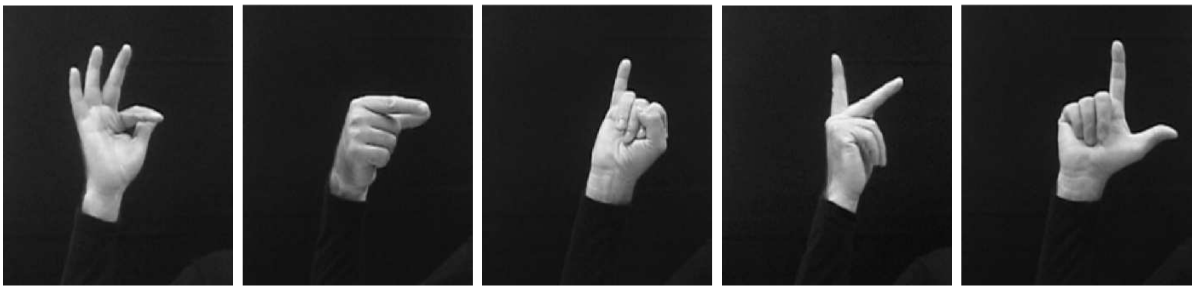
Afterwards, the action class initial and transition distributions $P(s_1|b)$ and $P(s_t|s_{t-1}, b)$ were learned separately for each behavior type (word) b, from specific training sequences. Similarly to our finger-counting application, these probability distributions were learned by counting the occurrences of starting classes and of transitions between classes from the training sequences (see Section 4.2.3). As mentioned in Section 4.2.3, one could alternatively estimate these parameters through an expectation-maximization (EM) approach, via the Baum-Welch algorithm (see [11, 120]).

4.3.5 Experimental Results

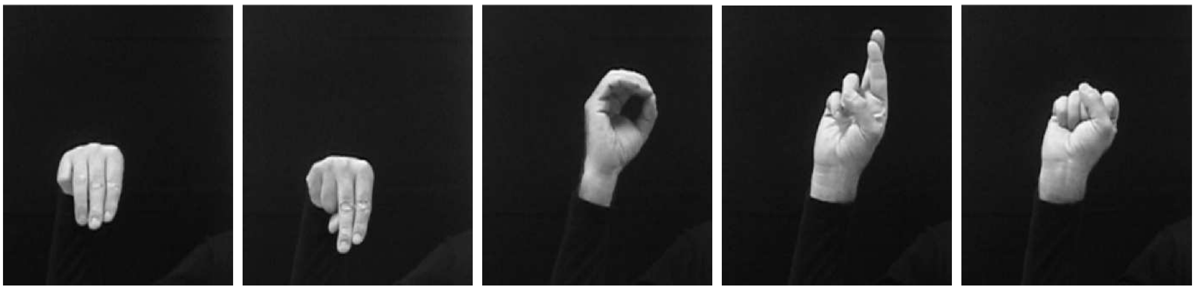
We tested the resulting implementation of our framework on image sequences of the same person finger-spelling words from the vocabulary. For testing, we have considered realistic conditions, involving a cluttered background, normal gesturing speed and changed lighting conditions with respect to the training image sequences. Despite the complexity of the task,



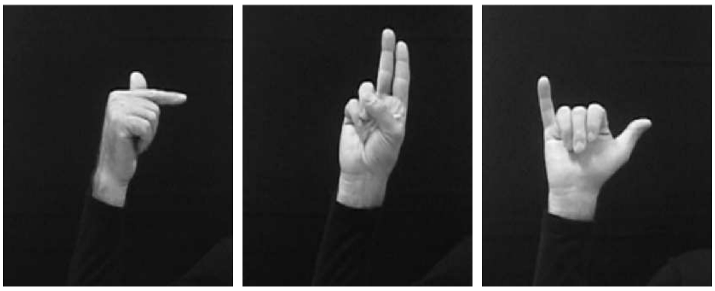
Letter A Letter B Letter C Letter D Letter E



Letter F Letter G Letter I Letter K Letter L



Letter M Letter N Letter O Letter R Letter S



Letter T Letter U Letter Y

Figure 4.6 — Sample images (and corresponding letter/action classes) from training sequences used in our application.

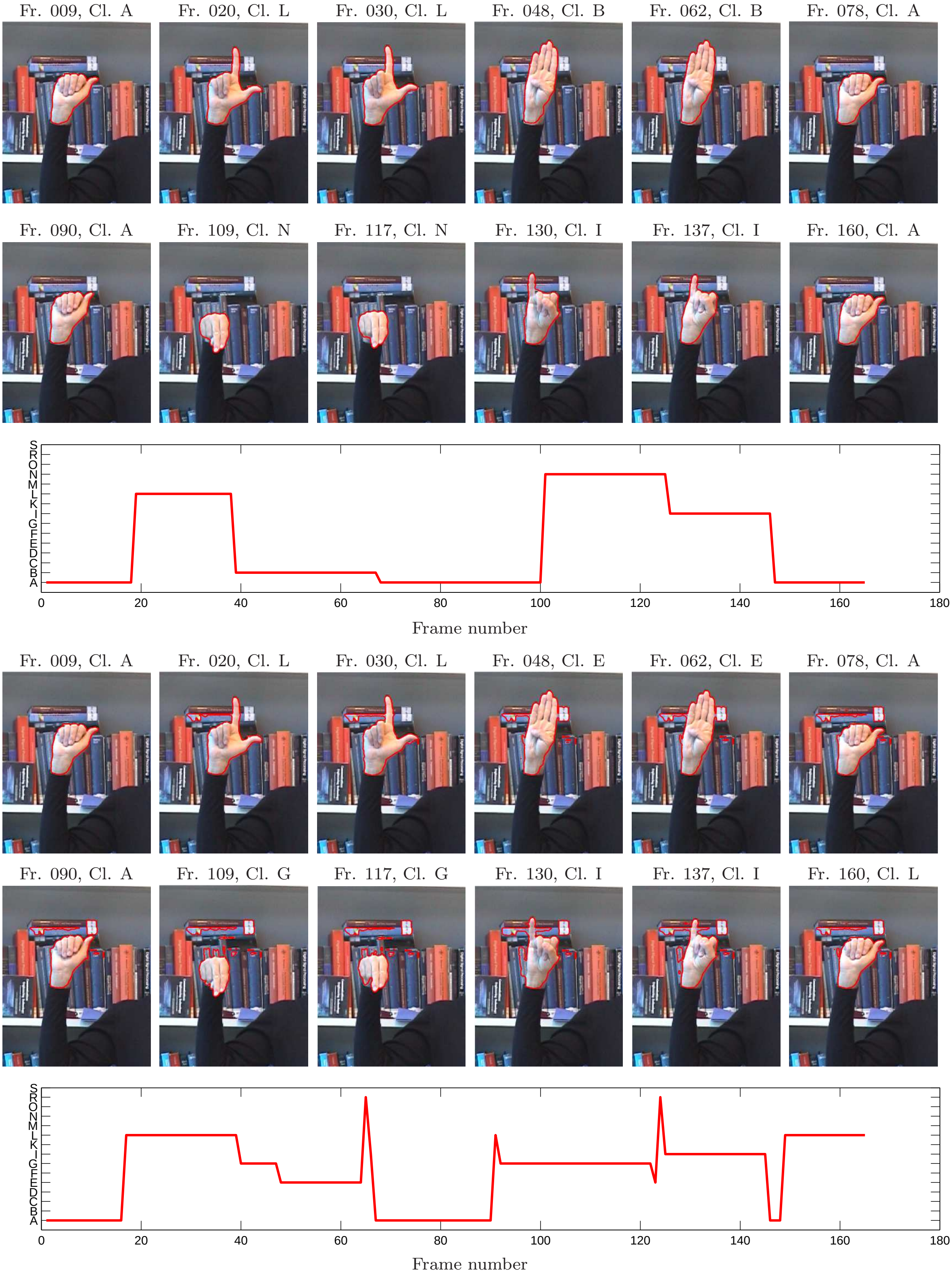


Figure 4.7 — Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Albania”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Algeria”.

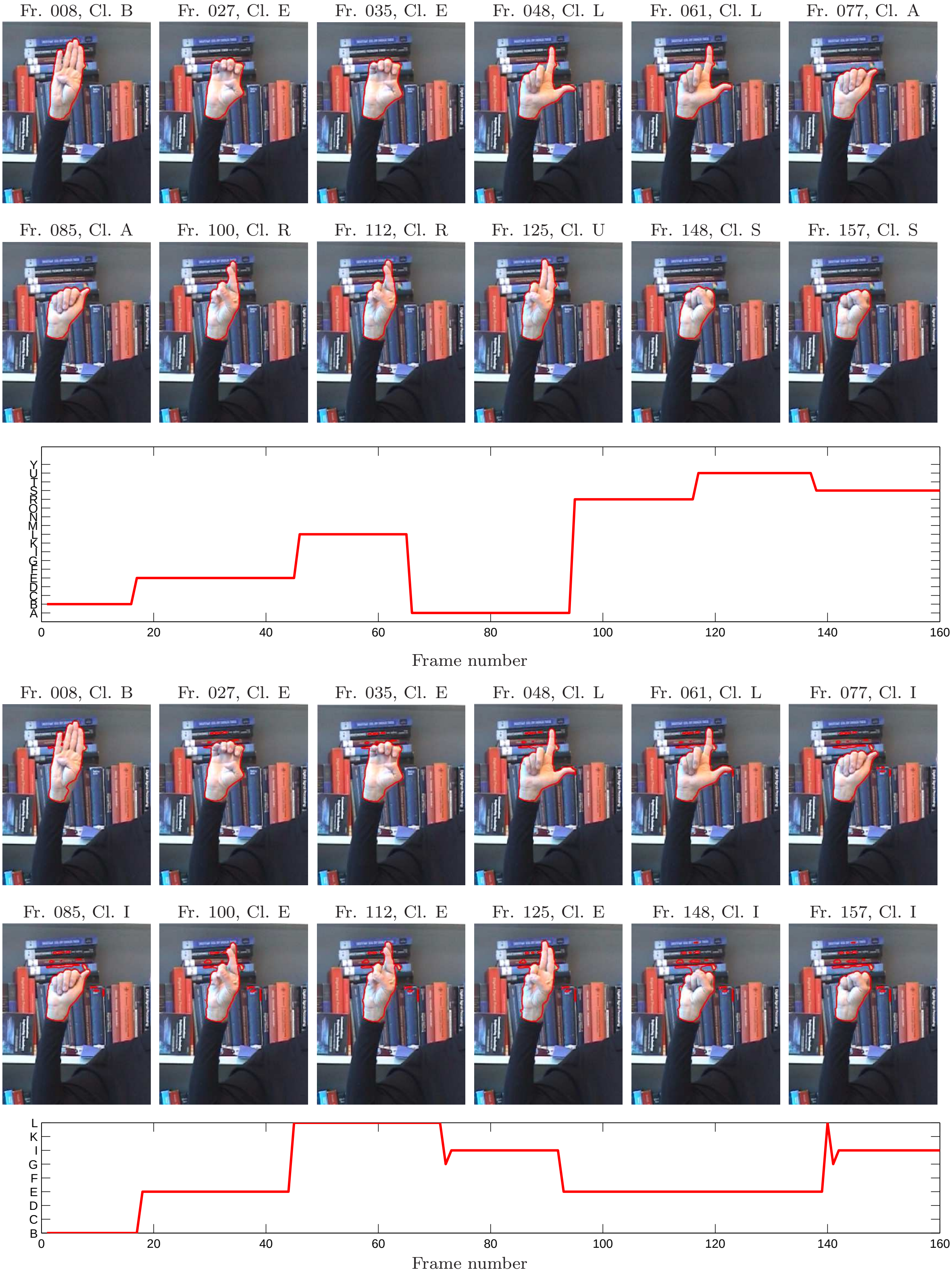


Figure 4.8 — Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word "Belarus". Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is "Belgium".

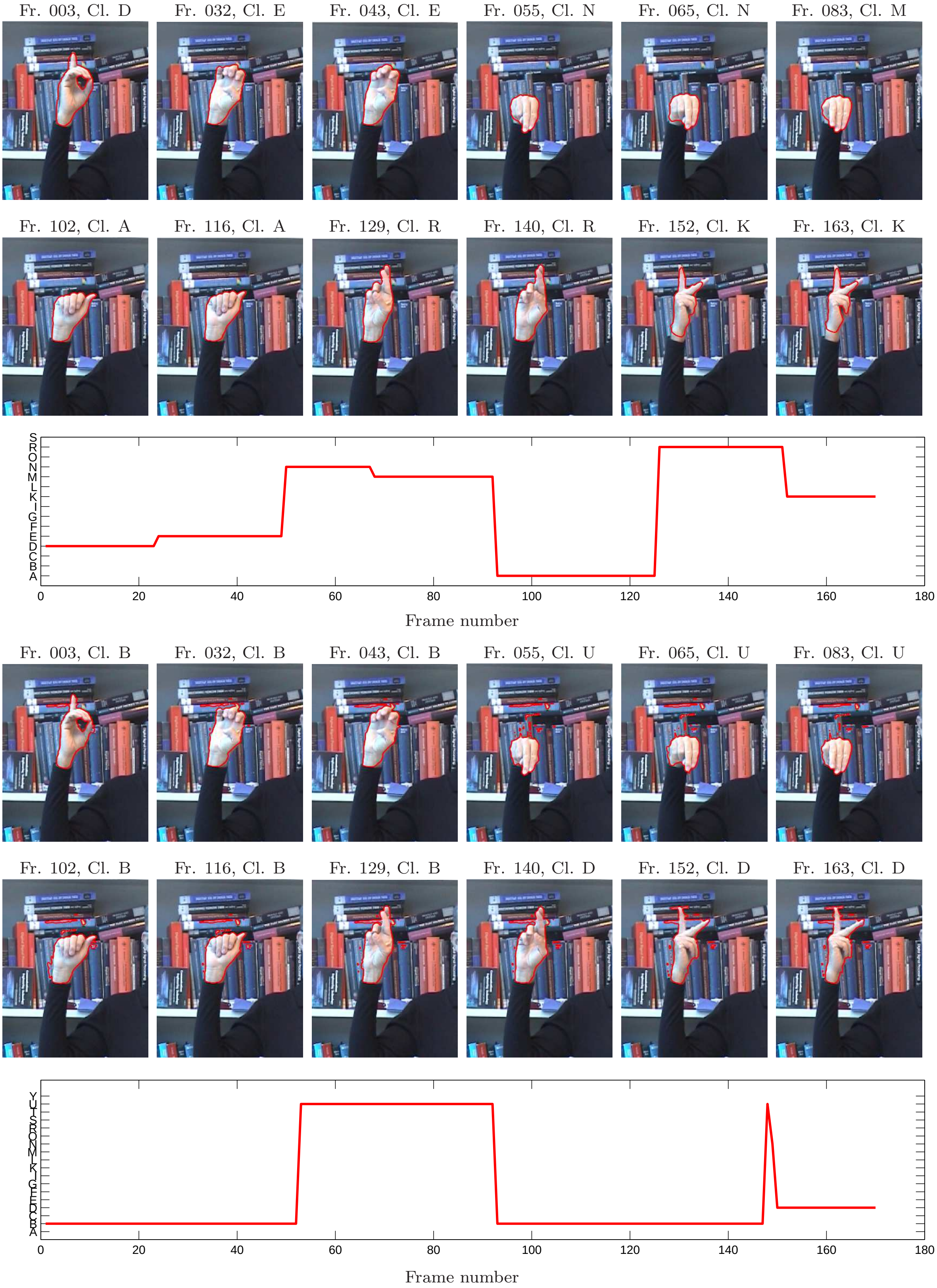


Figure 4.9 — Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word "Denmark". Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is "Burundi".

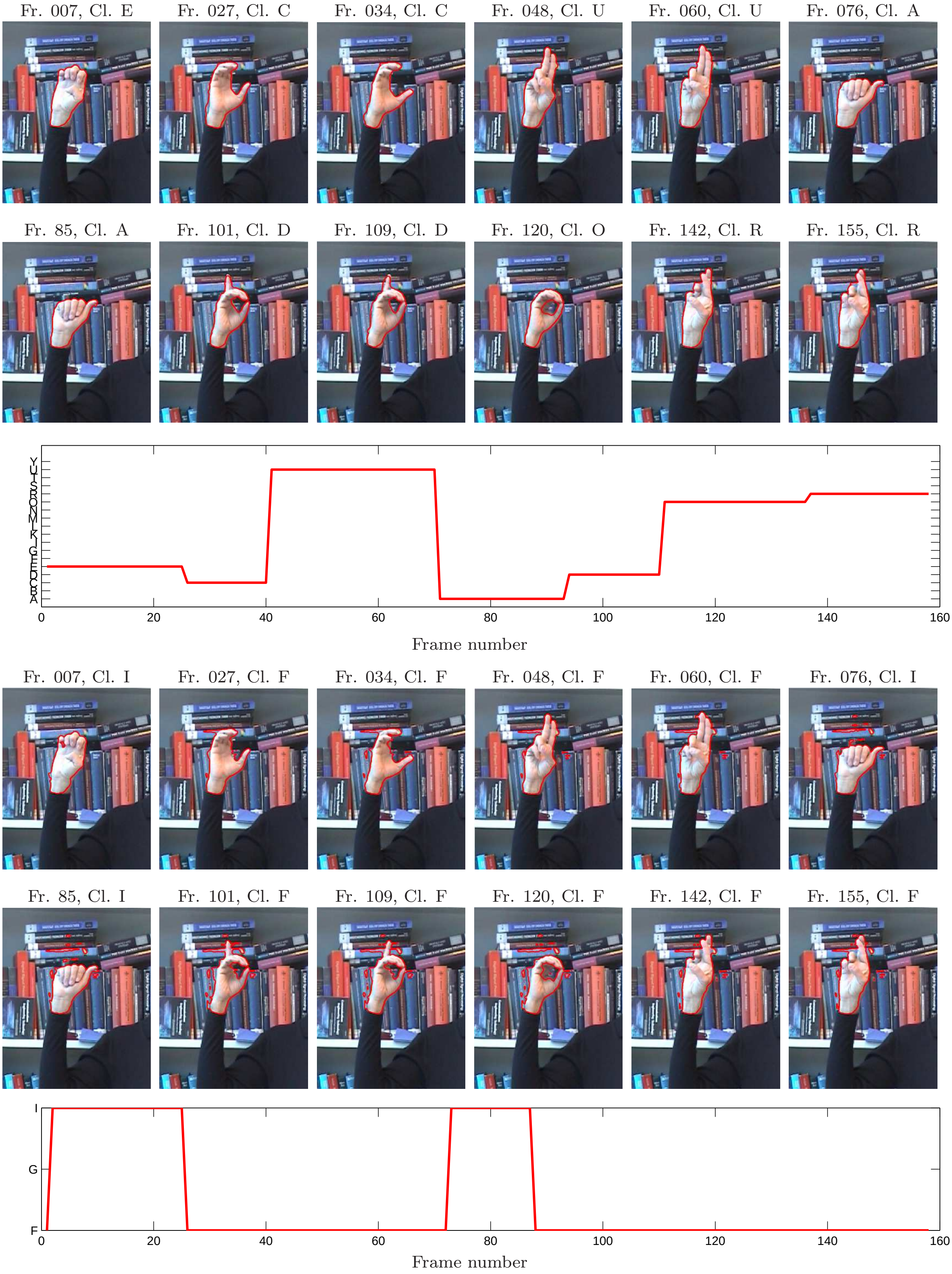


Figure 4.10 — Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word "Ecuador". Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is "Finland".

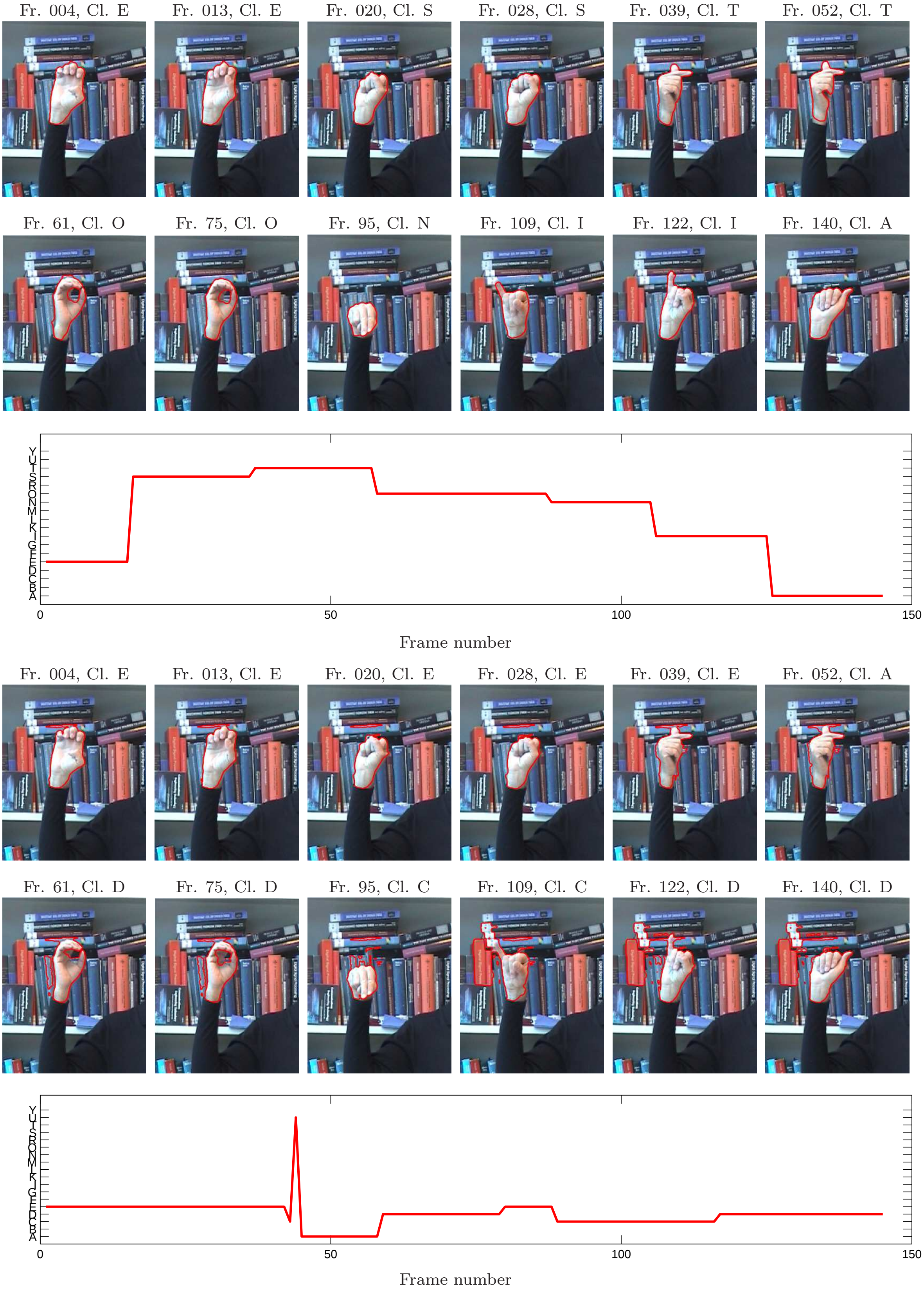


Figure 4.11 — Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word "Estonia". Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is "Ecuador".

the results are accurate in terms of the recognized words, due to the infusion of knowledge about the dynamics of vocabulary words via our collaborative framework.

In Figures 4.7, 4.8, 4.9, 4.10 and 4.11, rows 1 — 3, we present examples of collaborative segmentation and behavior recognition on five image sequences, which are correctly recognized by our framework as the words "Albania", "Belarus", "Denmark", "Ecuador" and "Estonia" respectively. The recognition framework helped orient segmentation towards the correct action classes at each time instance. Moreover, the dynamical PCA-based class prior models adapted to significant shape variations within behavior classes, allowing the segmentation of the hand in difficult cases of cluttered background. The frame-wise behavior recognition results for these sequences, yielded by backtracking for the winner behavior type, are presented in row 3 of each of these figures and correspond to our understanding of the sequences in terms of the executed gestures. In contrast, using the traditional (sequential) approach for recognition, i.e. first segmenting the image sequences (with the same variational approach, without prior models) and then performing recognition using the extracted contours (with the same Viterbi decoding scheme), produces completely erroneous results. Such results are presented for each of the above sequences, in Figs 4.7, 4.8, 4.9, 4.10 and 4.11, rows 4 — 6. In all these cases, the segmentation was side-tracked by the cluttered background, and as a result the sequences were miss-classified (as "Algeria", "Belgium", "Burundi", "Finland" and "Ecuador", respectively).

The variational segmentation parameters for the presented test sequences were $\alpha = 4000$, $\nu = 4000$, $\lambda_R = 1$, $\lambda_G = 0$ and $\lambda_B = 0$. The average execution time using un-optimized code (Matlab and C) was 6-7 minutes per frame. The segmenting contour of the first image of each sequence was determined by a rough manual initialization of the contour, followed by segmentation using only the image- and contour-based terms given by the piecewise-constant Chan-Vese model, adapted to color images:

$$\begin{aligned} E_{\text{image}}(I_1, \phi) + \nu E_{\text{contour}}(\phi) = \sum_{k \in \{R,G,B\}} \lambda_k \iint_\Omega (I_1^k - \mu_{\phi+}^k)^2 H(\phi) + (I_1^k - \mu_{\phi-}^k)^2 H(-\phi)\, dx\, dy \\ + \nu \iint_\Omega |\nabla H(\phi)|\, dx\, dy. \end{aligned} \tag{4.21}$$

As an alternative, our experience has shown that similarly good results can be obtained by using the following automatic segmentation method:

1. initialization with regularly distributed small circles,
2. variational segmentation with the piecewise-constant Chan-Vese model for color images (4.21),
3. elimination of small regions by morphological operations,
4. alignment of the mean level set functions $\overline{\phi}_i$ for each letter prior S_i with respect to the current contour, with genetic algorithms [51],

5. choice of the best fitting priors in terms of the distance:

$$d(\phi,\overline{\phi}_i)=\frac{\iint_\Omega \phi^2\,|\nabla\overline{\phi}_i|\,\delta(\overline{\phi}_i)\,dx\,dy}{\iint_\Omega |\nabla\overline{\phi}_i|\,\delta(\overline{\phi}_i)\,dx\,dy}, \tag{4.22}$$

6. variational segmentation using the image and contour terms (4.2) and the top 4 best fitting priors obtained at step 5, in a competition approach (prior term (3.37) with $w_1(S_i)=1$ and $P(A_1|s_1=S_i)=P_i(\phi)$, given by (4.18)).

This process is illustrated in Fig. 4.12 for the first image of the "Belarus" sequence.

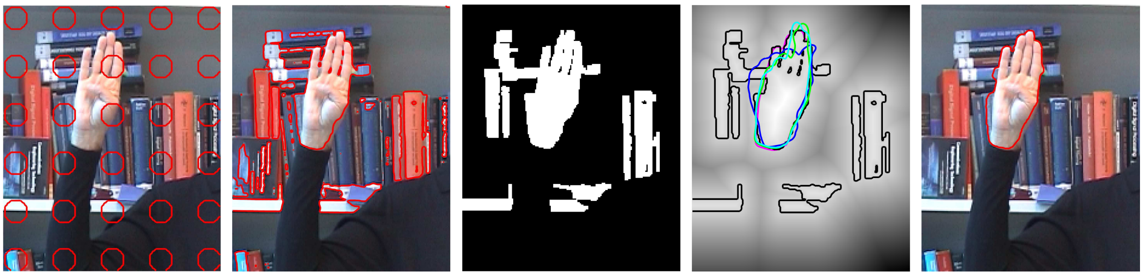

(a) (b) (c) (d) (e)

Figure 4.12 — Initialization process for the first frame of the "Belarus" sequence. (a) Initialization with small circles (step 1), (b) variational segmentation with the piecewise-constant Chan-Vese model for color images (step 2), (c) elimination of small regions by morphological operations (step 3): resulting binary mask, (d) alignment of the mean level set functions for the 4 top fitting priors (steps 4 and 5): image of the current level set function and its zero level set in black, together with the aligned means of the best fitting 4 priors (B, R, U, A) in color, (e) variational segmentation using the piecewise-constant Chan-Vese model for color images and the top 4 best fitting priors in a competition approach (step 6).

One of the advantages of performing behavior recognition (via the Viterbi decoding scheme) in collaboration with image segmentation is the fact that it offers us, at each instance t, the optimal classification of the sequence up to time t, which is used to guide further segmentation. This allows the correction of potential cases of miss-classification of previous frames, thus adding robustness to our approach. An example of miss-classification which is corrected in later frames is presented in Fig. 4.13, which shows partial classification results for the "Belarus" sequence. The partial classification result at frame 19 yields erroneous results (letter U instead of either B or E) for frames 17-19 , which are transition frames between two letters (see Fig. 4.13, first row). This result is corrected at frame 20, where letter E is clearly perceived and the Viterbi algorithm corrects the classification of the previous frames (Fig. 4.13, second row).

In order to show some limitations of the chosen implementation of our framework, in Fig. 4.14 we present two examples of miss-classification using our method. Rows 1 and 2 present the segmentation and recognition results of an image sequence representing the word "Belgium". This word is wrongfully classified as "Belgium". Examining the reasons for this decision, we note the similarity of the two words in terms of the contained letters — they have 4 common letters (B, E, L, U) in identical positions within the word — and also

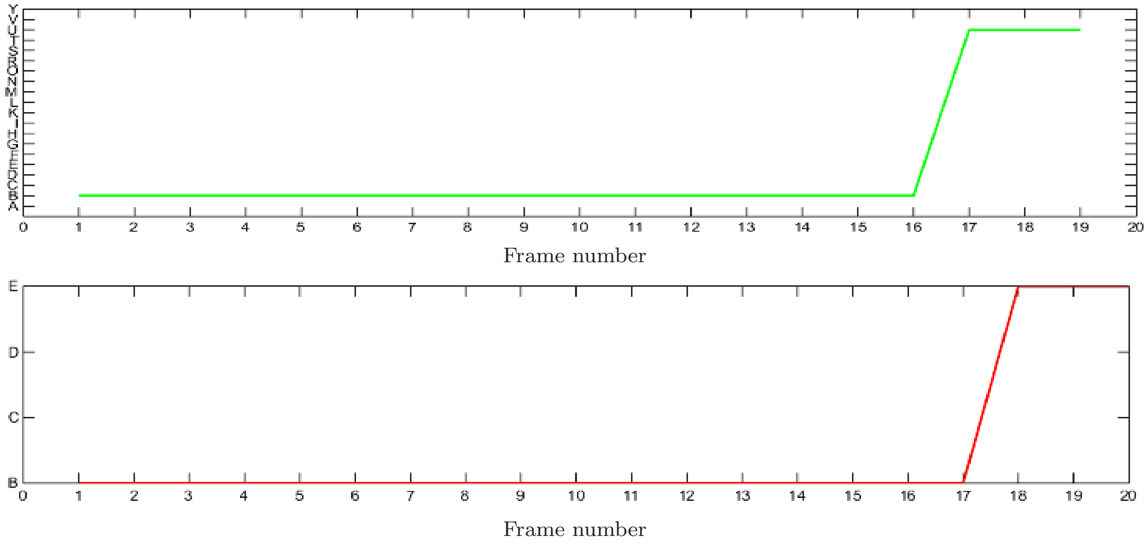


Figure 4.13 — Partial classification results for the "Belarus" sequence: at frame 19 (first row) and at frame 20 (second row). Mislabeling of 3 frames starting at frame 17 (first row), corrected in subsequent frames starting with 20 (second row).

in terms of the outlines of the rest of the letters (pairs (G, A) and (M, S)). Indeed, the first part of the word was correctly recognized as containing letters B, E, L. Further along, G was correctly segmented, but recognized as A, due the contour similarity between the two letters. Letter I was not correctly segmented due to the strong influence of the prior information, which was inclining towards the word "Belarus", due to the first letters recognized as B, E, L, A. Letter U was correctly segmented and recognized, being common to the two words and finally letter M, though correctly segmented, was recognized as S. Segmentation and recognition results for the second sequence, representing the word "Eritrea", are illustrated in rows 3 — 4 of Fig. 4.14. This sequence has been miss-classified as "Estonia". Similarly to the previous case, we note the three common letters of the two words: E, T and A. The first letter E was correctly segmented and classified, but the segmentation and recognition of the subsequent R was impaired by the strong prior information, imposing letter E. For similar reasons, the little finger differentiating I from S was not perceived. Common letter T was correctly segmented and classified, as expected. Then, during the transition from T to R, two frames were correctly segmented, and then classified as O and N, respectively. These are frames 88 and 89, illustrated in row 3 of Fig. 4.14. Indeed, the obtained contours of these transition frames resemble the contours of letters O and N from our training set (Fig. 4.6). Due to the amount of prior knowledge influence, the segmentation of the subsequent frames was affected and they were classified as letter A.

To interpret these results, we note that we performed our experiments by maintaining unchanged parameters for the segmentation of all images in all the test sequences. However, our experience has shown that improved results can be obtained by tuning these parameters to different test sequences. We did not consider such an approach, since it would render our method impractical to use. In the case of the above presented sequences, an important

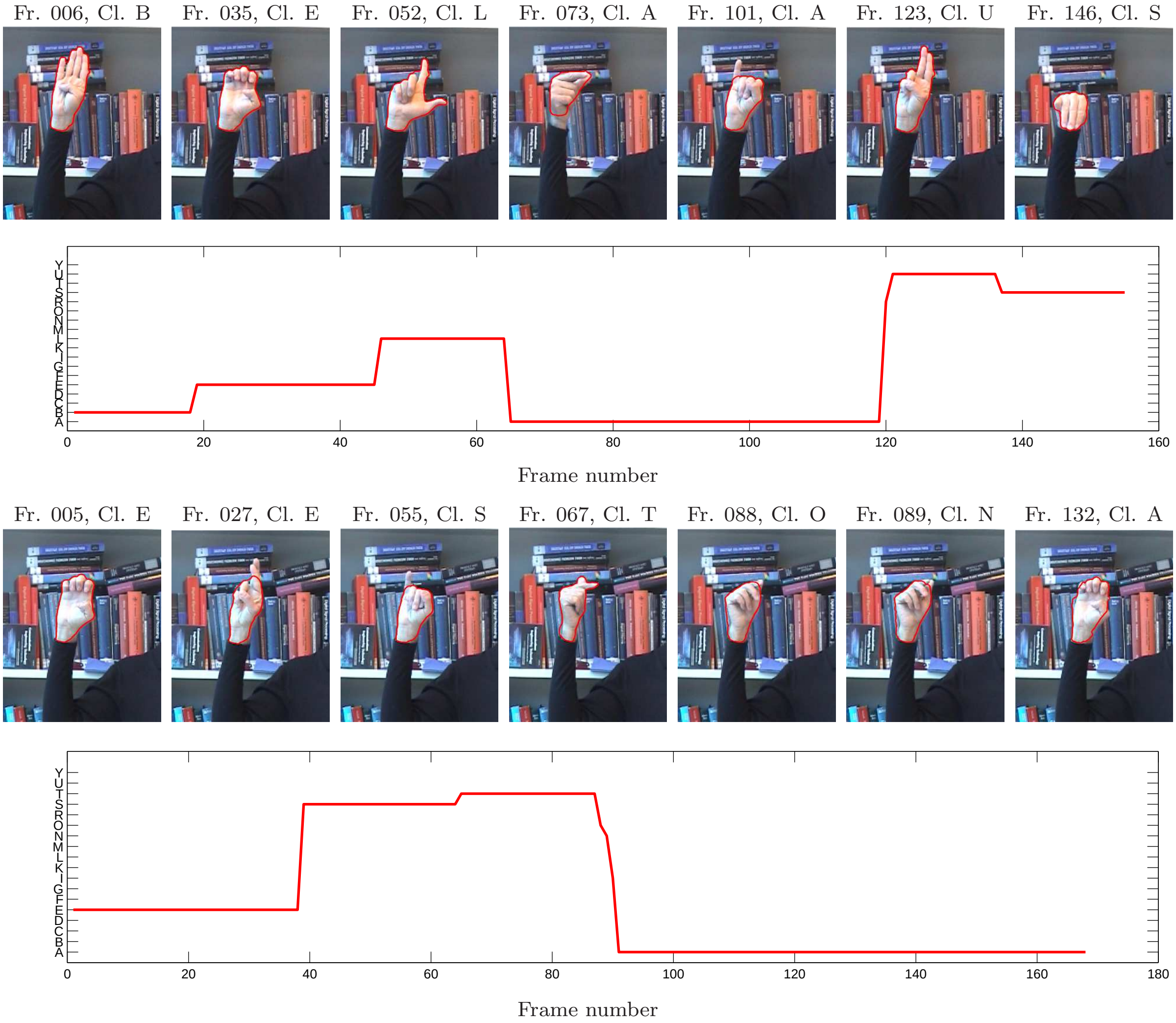


Figure 4.14 — Examples of erroneous classification using our method. Rows 1 — 2: segmentation and recognition of a sequence representing the word "Belgium", classified as "Belarus". Rows 3 — 4: segmentation and recognition of a sequence representing the word "Eritrea", classified as "Estonia".

factor for the failure of our method (beside the inherent similarity of the confounded words) is the misleading of the segmentation due to the too powerful influence of prior recognition information. The remedy for this problem would consist in slightly diminishing the weight α of the prior term in our segmentation energy. This would allow segmentation to better capture new letter characteristics, while receiving more moderate guidance from the recognition. For the reasons mentioned above, we did not consider such sequence-dependent parameter modifications.

Further analyzing the potential of the proposed framework implementation, we note that it can cope with difficult cluttered background, as shown by our experimental results. In this respect, it was shown to perform better than the traditional approach consisting of sequential segmentation and classification. However, its performance is still bounded due to the simplicity of the segmentation model. For instance, a challenging case for our method would be one where the average color levels of the background are similar to that of the hand. In this case, our method would be incapable of discriminating the hand from the background, despite prior knowledge regarding the most likely letter classes, offered by the recognition process. The solution lies in choosing more complicated segmentation models, (potentially involving histograms or texture information), which would in turn augment the computational costs of the method. Other challenges for our method would be poor resolution images (since it would increase class ambiguity in terms of hand contour), very noisy images (leading segmentation into unwanted local minima that match the wrong class prior information) or an important number of missing frames from the video sequences (misleading for the inference process). Moreover, an important factor for the success of our method is the clear spelling of the letters of each word, where the main letter durations are larger than the transitions between letters. A problematic situation for our approach would be given by very fast spelling, where the letter shapes would be undistinguishable due to co-articulation effects. This limitation could be partially overcome by training the method on such fast spelling sequences and/or inclusion of separate modeling for the transitions between letters.

Table 4.2 — Confusion matrix. Each row corresponds to the test sequences of one of the countries in our vocabulary (represented on the left of the row). The row entries for each column contain the percentage of these test sequences which were classified as belonging to the country associated with that column (represented on top of each column). The last column of the table gives the percentage of correctly classified test sequences for each country. The figure at the end of the last row represents the total percentage of correct classification over the ensemble of the test sequences.

Classification(%)	Albania	Algeria	Armenia	Austria	Belarus	Belgium	Burundi	Croatia	Denmark	Ecuador	Eritrea	Estonia	Finland	Georgia	Germany	Correct(%)
Albania	90	10	0	0	0	0	0	0	0	0	0	0	0	0	0	90
Algeria	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	100
Armenia	20	0	80	0	0	0	0	0	0	0	0	0	0	0	0	80
Austria	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	100
Belarus	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	100
Belgium	0	0	0	0	10	90	0	0	0	0	0	0	0	0	0	90
Burundi	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	100
Croatia	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	100
Denmark	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	100
Ecuador	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	100
Eritrea	0	0	0	0	0	0	0	0	0	0	50	50	0	0	0	50
Estonia	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	100
Finland	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	100
Georgia	10	50	0	10	0	0	0	0	0	0	0	0	0	30	0	30
Germany	0	20	0	30	0	0	0	0	0	0	0	0	0	0	40	40
Total(%)																85.3

To finish off the presentation of our experimental results, in Table 4.2 we illustrate the confusion matrix between the words in our vocabulary and the statistic recognition results per word and for the whole vocabulary. These results were obtained on a test sample of relatively limited size, which is ten words for each of the vocabulary words, obtained by courtesy of the Swiss Federation for the Hearing Impaired. As can be seen, for only 3 words out of 15 the results are quite poor ($\leq$ 50 %), mainly due to problems of parameter tuning, such as the ones exemplified above. However, for most words (12 out of 15), we obtain excellent recognition results (more than 80%), with a total recognition rate of 85.3 %.

4.4 Conclusion

In this chapter we presented two gesture recognition applications and their solutions in terms of our cooperative framework for segmentation and behavior recognition, developed in Chapter 3.

The first application concerns a finger-counting experiment involving four classes of hand gestures. For the discrimination of these gestures, the attribute we used is the hand contour, in a level set representation. We derived the solution to this application by choosing a particular implementation of our segmentation / recognition model presented in Section 3.3 of Chapter 3. To this end, we chose a Gaussian model for the class probabilities of the hand contour. Moreover, we instantiated the image- and contour-based terms of our segmentation model with the piecewise-constant Chan-Vese model. The training of our model was performed by segmentation of training image sequences, followed by the estimation of model parameters (mean and variance for the Gaussian class models, action class initial and transition distributions for our DBN) based on the extracted contours. The testing of the trained model on finger-counting image sequences featuring noisy images, a cluttered background and occlusions of the hand, yielded good segmentation and recognition results. This showed that the collaboration between image segmentation and behavior recognition renders our model robust against adverse imaging conditions.

The second application belongs to the area of sign-language recognition and regards the finger-spelling component of sign-language. Our target for recognition was a 15-word vocabulary based on the finger-spelling alphabet of the French-speaking region of Switzerland. A database of finger-spelling sequences of these words was obtained with the aid of the Swiss Federation for the Hearing Impaired. For the solution of the application, we used the extension of our framework for the recognition of a predefined behavior set, presented in Section 3.9 of Chapter 3. To this end, our vocabulary words were modeled as behavior types, while the action classes were given by the letters composing our vocabulary words. Similarly to the finger-counting application, we used a level set representation of the hand contour as attribute for the recognition task. To instantiate our framework, we chose a class probability model relying on a distance function with respect to a PCA-based class prior contour. For each letter class, these prior contours evolved during image segmentation in terms of their PCA coefficients, in order to match image characteristics. The evolution of the main segmentation contour, as well as the one of the prior contours, was based on the piecewise-constant Chan-Vese model. For the training of our model, slow-speed image sequences of our vocabulary words, finger-spelt in front of a simple background, were automatically segmented. Then PCA parameters were learned for each class from its corresponding extracted contours, after they were aligned for similarity transformations. Action class initial and transitions probabilities were learned separately for each word from corresponding training sequences. The trained model was tested on normal-speed finger-spelling sequences filmed in front of a cluttered background, which poses problems for regular feature extraction methods. A comparison with the traditional approach, where the segmentation and recognition phases are performed separately, shows the better performance of our collaborative approach.

5

Conclusion

The core issue that we have investigated in this thesis is the joining of two tasks that traditionally have been performed separately — image segmentation (for attribute extraction) and behavior recognition. The purpose of this union was to allow their collaboration towards improved results for both of them. In the following, we summarize the contributions of our study, discuss the limitations of the proposed methods and provide directions for future work.

5.1 Achievements

On the theoretical level, this thesis proposed a general framework for performing joint image segmentation and behavior recognition from image sequences. This framework was developed by formulating the double segmentation / recognition problem in terms of a Dynamic Bayesian Network, which incorporates a Hidden Markov Model and a generative image formation model. The solution to the problem was elaborated as a modified Viterbi decoding scheme, which blends recognition with segmentation along the image sequence. Guidelines and examples were provided regarding the choice of the free parameters of our framework, consisting mainly of modeling choices for the included probabilities, such as the attribute likelihood given the action class, the image probability given the object contour and attribute or the prior contour probability. Moreover, alternative learning methods for the parameters of the probability models were described.

In the context of our framework, a variational image segmentation model was proposed, as a natural derivation of the probabilistic segmentation formulation. This model is composed of a generic image segmentation term, including image- and contour-related constraints, and of a term which encapsulates a priori information about the attributes of the target object, offered by the recognition process. This term implements a competition between multiple priors stemming from several action classes, so that the final segmented object belongs to the most probable action class in light of image evidence and based on past experience accumulated by the recognition process.

Furthermore, the original framework was extended in order to allow the recognition of a predefined set of behavior types, each made up of a succession of simple actions, chosen from a finite set. A suitable Viterbi decoding scheme was proposed, in order to permit collaborative segmentation and behavior recognition in the new setting.

On a more practical level, two particular models implementing our general framework and its extension were proposed. These models were developed in order to solve two applications belonging to the field of gesture recognition. The first application is a finger-counting experiment involving four gesture classes and was solved via a particular model implementing our original general framework. The second application concerns finger-spelling recognition. The scope of this application was set to a vocabulary of 15 words, whose finger-spelling involved a number of 18 letter classes. The solution was provided via a model implementing the extension of our original framework for the recognition of a predefined behavior set. For both applications, the attribute used by recognition was the level set representation of the hand contour. Particular probabilistic models were chosen to fill in the optional parts of our framework. For both applications, the image- and contour-based segmentation terms were based on the piecewise-constant Chan-Vese model. For the finger-counting application, the attribute probability given the class was represented by a pixel-wise Gaussian model of the level set function. For the finger-spelling application, a class probability model based on a symmetric distance with respect to a PCA-represented class prior contour was chosen. Model training was detailed for both applications, following the guidelines provided in the general framework description. Testing of the two concrete models on image sequences from the two corresponding applications revealed their robustness with respect to difficult conditions, including noisy images, occlusions of the gesturing hand and cluttered background. Moreover, in the case of the finger-spelling application, a comparison with the traditional approach, which separates attribute extraction (via segmentation) and behavior recognition, showed the better performance of our collaborative model in terms of both segmentation and recognition results.

5.2 Discussion and Future Work

In the formulation of our general segmentation / recognition framework, we regard behavior as a succession of simple actions. We describe these actions in terms of object attributes that are emitted with a certain probability given a particular action class. Furthermore, we characterize the succession of action classes by a Markov chain. This kind of behavior description corresponds to a Hidden Markov Model. The temporal dependency between successive attributes is modeled in terms of transition probabilities between the discrete hidden action classes that produce the attributes. As future work, we can imagine extensions of our model in order to incorporate more complex temporal dependencies between attributes. One example could be the inclusion of a auto-regressive dependency between successive attributes, whose parameters would depend on the action class. Such an extension would facilitate the application of the model to scenarios where the attributes are changing continuously, but in a predictable way depending on the action class, for example when wishing to discriminate between activities like walking and running.

Let us now look at the application of our framework to applications such as the finger-spelling recognition one. As our experimental results have shown, our model is able to cope with important amounts of background clutter due to the infusion of prior knowledge from the recognition process. Nevertheless, our proposed model can still become sidetracked from the correct segmentation if the objects in the background are too similar in average color with respect to the hand. This aspect could be improved by the incorporation of more complex image-based segmentation models, including more complex models of color (e.g. histogram-based), texture models, or the use of a piecewise-smooth formulation instead of the piecewise-constant model that we have employed. Another idea would be to incorporate a form of background modeling, which could be rendered adaptive in time, so as not to constrain the application to a fixed background. However, we should mention that our choice of a rather simplistic model was partly motivated by considerations regarding computation time, which would augment with the use of more complicated models.

Indeed, computation time is one of the sensitive points of our framework. This is mainly due to the fact that it relies on a variational method for image segmentation. The numerous advantages of variational segmentation methods, among which the rigorous mathematical formulation and the flexible inclusion of various criteria, were explained in Chapter 2. However, the typical numerical implementations of these methods require the iterative evolution of the segmentation contour until convergence, using evolution time steps which are limited in size by considerations regarding the stability of the numerical schemes. This translates into relatively long computation times per image. In our case, to speed up computation, we used the narrow-band method [1] for updating the level set function representing our segmentation contour, and the fast-marching method [1] for the re-initialization of the level set function to a signed distance function. However, additional computation time could be gained by considering a multi-grid numerical implementation. Another option would be to replace the level set contour representation by a B-spline parametric one, which would drastically reduce the dimensions of our problem. However, we would loose the ability to capture interior object contours (without additional complications), which come up for instance in representing the hand contour for letter O in our finger-spelling application. Last, but not least, the optimization of the code could be considered, by a C-only implementation and processor optimizations.

Regarding the testing of our framework, it would be interesting to extend our applications (in particular the finger-spelling one) to several gesturing persons, and also to extend the testing scenarios to different background and lighting configurations, as well as cases of missing frames from the test image sequences.

Appendix A

A.1 The Minimization of Functionals Using the Calculus of Variations and Gradient Descent

In the following, we briefly outline a classical method used for the minimization of typical functionals encountered in image processing problems. This method is based on the calculus of variations and gradient descent (cf. [130]).

We begin by presenting the one-dimensional (1D) case. Given a 1D function $u(x) : [0,1] \longrightarrow \mathbb{R}$, we wish to minimize a given energy functional

$$E(u) = \int_0^1 F(u, u')dx, \tag{A.1}$$

subject to given boundary conditions $u(0) = a$ and $u(1) = b$. Here $F : \mathbb{R}^2 \longrightarrow \mathbb{R}$ is dictated by the particular application to solve and depends on the function u and on its derivative u'.

In classical calculus, the extrema of a *function* $f(x) : \mathbb{R} \longrightarrow \mathbb{R}$ are reached in those points of the domain where $f'(x) = 0$. Likewise, in the calculus of variations we can attain the extrema of the *functional* $E(u)$ in those points where $E' = 0$, where $E' = \frac{\partial E}{\partial u}$ is the first variation of $E(u)$. As shown in [130], this leads to the following necessary condition in order for u to be an extremum of $E(u)$:

$$\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) = 0. \tag{A.2}$$

This is the Euler-Lagrange equation for the 1D case. Similarly, for an energy of the form

$$E(u) = \int_0^1 F(u, u', u'')dx, \tag{A.3}$$

the Euler-Lagrange equation is given by

$$\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) + \frac{d^2}{dx^2}\left(\frac{\partial F}{\partial u''}\right) = 0. \tag{A.4}$$

For the 2D case, the equations are analogous. Given a function $u(x, y) : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$, we wish to minimize the following energy with respect to u:

$$E(u) = \iint_\Omega F\,(u,\, u_x,\, u_y,\, u_{xx},\, u_{yy})\; dx\, dy. \tag{A.5}$$

The necessary condition for u to be an extremum point for E is given by the Euler-Lagrange equation:

$$\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u_x}\right) - \frac{d}{dy}\left(\frac{\partial F}{\partial u_y}\right) + \frac{d^2}{dx^2}\left(\frac{\partial F}{\partial u_{xx}}\right) + \frac{d^2}{dy^2}\left(\frac{\partial F}{\partial u_{yy}}\right) = 0. \tag{A.6}$$

The remaining problem now is finding a solution for the Euler-Lagrange equation, that we denote by

$$L(u) = 0,$$

where $L(u)$ designates the left-hand side of equations such as (A.6). Generally, in image processing tasks this equation is impossible to solve analytically. Therefore, numerical solutions are usually preferred. One of the most commonly used methods is the *gradient descent*. The basic idea is that in order to find a solution for $L(u) = 0$, we numerically solve the PDE

$$\frac{\partial u}{\partial t} = L(u), \tag{A.7}$$

starting from the initial condition $u(0) = u_0$, where u_0 is the given initial data and t is an artificial time-marching parameter. Once we reach the steady state of this equation, that is, when

$$\frac{\partial u}{\partial t} = 0, \tag{A.8}$$

then we have found the solution $u^* = u$ to the Euler-Lagrange equation:

$$L(u^*) = 0.$$

This gradient descent method is not guaranteed to reach the optimal solution. If the energy to minimize is not convex, the solution to the PDE (A.7) may not be unique or may vary depending on the initial condition which is used. Its use is nonetheless widespread, since in many cases a local minimum of the energy functional constitutes an acceptable solution to the given problem.

A.2 Image Segmentation Using the Gaussian Prior Model, for the Finger-counting Application

A.2.1 Evolution Equations

Let us denote $\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)) = \phi_i(h_{\boldsymbol{T}^i}(x,y))/s^i$. Then, the evolution equation for the segmenting contour ϕ is given by:

$$\begin{aligned}\frac{\partial \phi}{\partial t}(x,y) = {} & \delta_\varepsilon(\phi(x,y))\Bigg((I(x,y)-\mu_-)^2 - (I(x,y)-\mu_+)^2 \\ & + \nu\,\mathrm{div}\bigg(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\bigg)\Bigg) + \alpha\sum_{i=1}^{M}\frac{\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)) - \phi(x,y)}{\sigma_i(h_{\boldsymbol{T}^i}(x,y))}\,L_i^2,\end{aligned} \tag{A.9}$$

where δ_ε is a regularized version of the Dirac function:

$$\delta_\varepsilon(x) = \frac{\varepsilon}{\pi(x^2+\varepsilon^2)}. \tag{A.10}$$

The similarity transformation parameters of each prior evolve according to:

$$\begin{aligned}\frac{\partial \tau^i}{\partial t} = & -\iint_\Omega \frac{1}{\sigma_i(h_{\boldsymbol{T}^i}(x,y))}\Big(\nabla\sigma_i(h_{\boldsymbol{T}^i}(x,y))\cdot\frac{\partial}{\partial\tau^i}(h_{\boldsymbol{T}^i}(x,y))\Big)\,dx\,dy \\ & + \iint_\Omega \frac{\phi(x,y)-\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))}{\sigma_i^2(h_{\boldsymbol{T}^i}(x,y))}\frac{\partial}{\partial\tau^i}(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))\,dx\,dy \\ & + \iint_\Omega \frac{(\phi(x,y)-\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))^2}{\sigma_i^3(h_{\boldsymbol{T}^i}(x,y))}\Big(\nabla\sigma_i(h_{\boldsymbol{T}^i}(x,y))\cdot\frac{\partial}{\partial\tau^i}(h_{\boldsymbol{T}^i}(x,y))\Big)\,dx\,dy\end{aligned} \tag{A.11}$$

where τ^i stands for each of s^i, θ^i, and T^i. Moreover,

$$\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))) = \nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))\cdot\frac{\partial}{\partial\tau_i}(h_{\boldsymbol{T}^i}(x,y)), \tag{A.12}$$

if $\tau^i = \theta^i$, T^i and

$$\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))) = \nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))\cdot\frac{\partial}{\partial\tau_i}(h_{\boldsymbol{T}^i}(x,y)) - \frac{1}{s^i}\,\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)), \tag{A.13}$$

if $\tau^i = s^i$. The derivatives $\partial(h_{\boldsymbol{T}^i}(x,y))/\partial\tau^i$ are computed as follows:

$$\frac{\partial}{\partial s}(h_{\boldsymbol{T}}(x,y)) = \begin{pmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{pmatrix}\begin{bmatrix}x \\ y\end{bmatrix}, \tag{A.14}$$

$$\frac{\partial}{\partial\theta}(h_{\boldsymbol{T}}(x,y)) = s\begin{pmatrix}-\sin\theta & \cos\theta \\ -\cos\theta & -\sin\theta\end{pmatrix}\begin{bmatrix}x \\ y\end{bmatrix}, \tag{A.15}$$

$$\frac{\partial}{\partial T_x}(h_{\boldsymbol{T}}(x,y)) = \begin{bmatrix}1 \\ 0\end{bmatrix}, \tag{A.16}$$

$$\frac{\partial}{\partial T_y}(h_{\boldsymbol{T}}(x,y)) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{A.17}$$

The labels L_i, $i = 1..M$ evolve according to:

$$\begin{aligned}\frac{\partial L_i}{\partial t} =& L_i\Bigg(\log w_t(S_i) - \iint_\Omega \Bigg(\frac{(\phi(x,y) - \hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))^2}{2\sigma_i^2(h_{\boldsymbol{T}^i}(x,y))} \\ &+ \log \sigma_i(h_{\boldsymbol{T}^i}(x,y))\Bigg)\, dx\, dy \, + 2\beta\left(1 - \sum_{i=1}^{M} L_i^2\right)\Bigg).\end{aligned} \tag{A.18}$$

The update equation for the Lagrange multiplier β is as follows:

$$\beta = \frac{\sum_{i=1}^{M} L_i^2 \log\left(w_t(S_i)P_i(\phi)\right)}{2\sum_{i=1}^{M} L_i^2\left(\sum_{i=1}^{M} L_i^2 - 1\right)}, \tag{A.19}$$

with $P_i(\phi)$ given by (4.4).

A.2.2 Numerical Approach

To minimize the total energy (3.35), with $E_{\text{image}}(I_t, C_t) + \nu E_{\text{contour}}(C_t)$ given by (4.2) and E_{prior} given by (4.6), we use the evolution equations (A.9), (A.11) and (A.18). We solve these equations numerically by iterating the following steps until convergence is reached:

1. Computation of the mean intensities μ_+ and μ_- over image I regions corresponding to the positive, respectively negative regions of the level set function ϕ.

2. Computation of the class prior information $\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))$ and $\sigma_i(h_{\boldsymbol{T}^i}(x,y))$ from the average LSF $\phi_i(x,y)$ and the variance $\sigma_i(x,y)$, by applying the similarity transformations $h_{\boldsymbol{T}^i}$ (4.5) via the B-splines interpolation method [146].

3. Computation of the curvature $\operatorname{div}(\nabla\phi(x,y)/|\nabla\phi(x,y)|)$ and of the gradients $\nabla\sigma_i(h_{\boldsymbol{T}^i}(x,y))$ and $\nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))$ using a central difference scheme.

4. Calculation of the temporal derivatives in (A.9), (A.11) and (A.18) using a forward difference approximation.

5. Re-distancing of the level set function ϕ to a signed distance function, using the fast marching method of [1].

6. Update of the Lagrange multiplier β according to (A.19).

A.3 Image Segmentation Using the PCA-based Prior Model, for the Finger-spelling Application

A.3.1 Evolution Equations

The evolution of the main contour, given by the level set function ϕ, is governed by the equation:

$$\begin{aligned}\frac{\partial\phi(x,y)}{\partial t} =& \,\delta_\varepsilon(\phi(x,y))\Bigg((I(x,y)-\mu_-)^2-(I(x,y)-\mu_+)^2\Bigg)+\nu\,\mathrm{div}\bigg(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\bigg)\\ &+\alpha\sum_{i=1}^{M}\Bigg(\hat{\phi}_i^2(h_{\boldsymbol{\tau}^i}(x,y))\mathrm{div}\bigg(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\bigg)\delta_\varepsilon(\phi(x,y))\\ &+\Bigg(\nabla\hat{\phi}_i^2(h_{\boldsymbol{\tau}^i}(x,y))\cdot\bigg(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\bigg)\Bigg)\,\delta_\varepsilon(\phi(x,y))\\ &-2\phi(x,y)|\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))|\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\Bigg)L_i^2.\end{aligned} \tag{A.20}$$

The similarity transformation parameters of each prior evolve according to:

$$\begin{aligned}\frac{\partial\tau^i}{\partial t} =& \iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i-})^2\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\,dx\,dy\\ &-\iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i+})^2\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\,dx\,dy\\ &-2\iint_\Omega \hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))|\nabla\phi(x,y)|\delta_\varepsilon(\phi(x,y))\,dx\,dy\\ &-\iint_\Omega \phi^2(x,y)\Bigg(|\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)|\delta'_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\\ &+\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\frac{1}{|\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)|}\Bigg((\hat{\phi}_i)_x(h_{\boldsymbol{\tau}^i}(x,y))\frac{\partial}{\partial\tau_i}((\hat{\phi}_i)_x(h_{\boldsymbol{\tau}^i}(x,y))))+\\ &(\hat{\phi}_i)_y(h_{\boldsymbol{\tau}^i}(x,y)\frac{\partial}{\partial\tau_i}((\hat{\phi}_i)_y(h_{\boldsymbol{\tau}^i}(x,y))\Bigg)\Bigg)\,dx\,dy,\end{aligned} \tag{A.21}$$

where τ^i stands for each of s^i, θ^i, and T^i, and $(\hat{\phi}_i)_x$, $(\hat{\phi}_i)_y$ are the x and y derivatives of $\hat{\phi}_i$. The derivatives $\partial\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))/\partial\tau_i$, $\partial(\hat{\phi}_i)_x(h_{\boldsymbol{\tau}^i}(x,y))/\partial\tau_i$ and $\partial(\hat{\phi}_i)_y(h_{\boldsymbol{\tau}^i}(x,y))/\partial\tau_i$ are computed as in (A.12), (A.13).

The evolution equation for the j^{th} PCA coefficient of prior class S_i is:

$$\begin{aligned}\frac{\partial \mathbf{c}^i_j}{\partial t} =& \frac{1}{s^i}\iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i -})^2 \delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))E_{ij}(h_{\boldsymbol{T}^i}(x,y))\,dx\,dy \\ &-\frac{1}{s^i}\iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i +})^2 \delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))E_{ij}(h_{\boldsymbol{T}^i}(x,y))\,dx\,dy \\ &-\frac{2}{s^i}\iint_\Omega \hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))E_{ij}(h_{\boldsymbol{T}^i}(x,y))|\nabla\phi(x,y)|\delta_\varepsilon(\phi(x,y))\,dx\,dy \\ &-\frac{1}{s^i}\iint_\Omega \phi^2(x,y)\Bigg(|\nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))|\delta'_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))E_{ij}(h_{\boldsymbol{T}^i}(x,y)) \\ &+\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))\frac{1}{|\nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))|}\Bigg((\hat{\phi}_i)_x(h_{\boldsymbol{T}^i}(x,y))(E_{ij})_x(h_{\boldsymbol{T}^i}(x,y)) \\ &+(\hat{\phi}_i)_y(h_{\boldsymbol{T}^i}(x,y))(E_{ij})_y(h_{\boldsymbol{T}^i}(x,y))\Bigg)\Bigg)\,dx\,dy,\end{aligned} \tag{A.22}$$

where E_{ij} is the j^{th} eigenvector of class S_i, arranged as the columns of an image-sized matrix (continuously interpolated) and $(E_{ij})_x$ and $(E_{ij})_y$ are its x and y derivatives, respectively.

The labels L_i, $i = 1..M$, evolve according to:

$$\begin{aligned}\frac{\partial L_i}{\partial t} =& L_i\Bigg(\log w_t(S_i) - \iint_\Omega \hat{\phi}_i^{\,2}(h_{\boldsymbol{T}^i}(x,y))|\nabla\phi(x,y)|\delta_\varepsilon(\phi(x,y))\,dx\,dy \\ &-\iint_\Omega \phi^2(x,y))|\nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))|\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))\,dx\,dy \\ &+2\beta\left(1-\sum_{i=1}^{M} L_i^2\right)\Bigg).\end{aligned} \tag{A.23}$$

The update equation for the Lagrange multiplier β is as follows:

$$\beta = \frac{\sum_{i=1}^M L_i^2 \log\left(w_t(S_i)P_i(\phi)\right)}{2\sum_{i=1}^M L_i^2\left(\sum_{i=1}^M L_i^2 - 1\right)}, \tag{A.24}$$

with $P_i(\phi)$ given by (4.18).

A.3.2 Numerical Approach

To minimize energy (3.35), with $E_{\text{image}}(I_t, C_t) + \nu E_{\text{contour}}(C_t)$ given by (4.19) and E_{prior} given by (4.20), we use the evolution equations (A.20), (A.21), (A.22) and (A.23). We solve these equations numerically by iterating the following steps until convergence is reached:

1. Computation of the mean intensities μ_+ and μ_- over image I regions corresponding to the positive, respectively negative regions of the level set function ϕ.

2. Computation of the mean intensities $\mu_{\hat{\phi}_i +}$ and $\mu_{\hat{\phi}_i -}$ over image I regions corresponding to the positive, respectively negative regions of the level set functions $\hat{\phi}_i$.

3. Computation of the class prior information $\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y)))$ and $E_{ij}(h_{\boldsymbol{T}^i}(x,y))$ from the average LSF $\phi_i(x,y)$ and the eigenvectors $E_{ij}(x,y)$, by using (4.16) and applying the similarity transformations $h_{\boldsymbol{T}^i}$ (4.5) via the B-splines interpolation method [146].

4. Computation of the curvature $\mathrm{div}(\nabla\phi(x,y)/|\nabla\phi(x,y)|)$, derivatives $(\hat{\phi}_i)_x$, $(\hat{\phi}_i)_y$, $(E_{ij})_x$ and $(E_{ij})_y$ and gradients $\nabla\phi(x,y)$ and $\nabla\hat{\phi}_i(h_{\boldsymbol{T}^i}(x,y))$ using a central difference scheme.

5. Calculation of the temporal derivatives in (A.20), (A.21), (A.22) and (A.23) using a forward difference approximation.

6. Re-distancing of the level set function ϕ with the fast marching method of [1].

7. Update of the Lagrange multiplier β according to (A.24).

Bibliography

[1] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1995.

[2] J.K. Aggarwal and S. Park. Human motion: modeling and recognition of actions and interactions. In *Proceedings of the Second International Symposium on 3D Data Processing, Visualization and Transmission*, Thessaloniki, Greece, September 2004.

[3] M. Ahmad and S.W. Lee. Human action recognition using multi-view image sequences features. In *International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, April 2006.

[4] L. Ambrosio. A compactness theorem for a special class of functions of bounded variation. *Bolletino Della Unione Matematica Italiana*, 3-B:857–881, 1989.

[5] L. Ambrosio. Variational problems in sbv and image segmentation. *Acta Applicandae Mathematicae*, 17:1–40, 1989.

[6] L. Ambrosio. Existence theory for a new class of variational problem. *Arch. Rational Mech. Anal.*, 111:291–322, 1990.

[7] L. Ambrosio and V.M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via γ-convergence. *Communications on Pure and Applied Mathematics,*, 43:999–1036, 1991.

[8] L. Ambrosio, N. Fusco, and D. Pallara. Partial regularity of free discontinuity sets, II. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze Sér. 4*, 24(1): 39–62, 1997.

[9] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition)*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2006.

[10] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients. *SIAM Journal on Applied Mathematics*, 63(6):2128–2154, 2003.

[11] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[12] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:69–77, 2004.

[13] H. Birk, T.B. Moeslund, and C.B. Madsen. Real-time recognition of hand alphabet gestures using principal component analysis. In *Proceedings of Scandinavian Conference on Image Analaysis*, pages 261–268, 1997.

[14] J. Blackburn and E. Ribeiro. Human motion recognition using isomap and dynamic time warping. In *ICCV Workshop on Human Motion, in Lecture Notes in Computer Science*, pages 285–298, Rio de Janeiro, Brazil, 2007.

[15] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1997.

[16] A. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London*, 352:1257–1265, 1997.

[17] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3): 257–267, 2001.

[18] A. Bonnet. Sur la régularité des bords des minima de la fonctionelle de mumford-shah. *Comptes rendus de l'Académie des sciences Paris, Série I*, 321(9):1275–1279, 1995.

[19] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *The 8th European Conference on Computer Vision*, pages 391–401, 2004.

[20] A. Braides. Approximation of free-discontinuity problems. In *Lecture Notes in Mathematics, Springer-Verlag*, volume 1694, 1998.

[21] X. Bresson. *Image segmentation with variational active contours*. PhD thesis, Swiss Federal Insitute of Technology (EPFL), Lausanne, Switzerland, 2005.

[22] X. Bresson, P. Vandergheynst, and J.P. Thiran. A priori information in image segmentation: energy functional based on shape statistical model and image information. In *IEEE International Conference on Image Processing (ICIP)*, pages 425–428, 2003.

[23] X. Bresson, P. Vandergheynst, and J.P. Thiran. A variational model for object segmentation using boundary information and shape prior driven by the Mumford-Shah functional. *International Journal of Computer Vision*, 28(2):145 – 162, July 2006.

[24] X. Bresson, S. Esedoglu, P. Vandergheynst, J.P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007.

[25] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In F. Sgallari, A. Murli, and N. Paragios, editors, *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*, volume 4485 of *LNCS*, pages 203–213, Ischia, Italy, May 2007. Springer.

[26] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[27] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Journal of Computational Physics*, 79:12–49, 1988.

[28] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 694–699, Boston, USA, 1995.

[29] A. Chambolle. Image segmentation by variational methods: Mumford and Shah functional and the discrete approximation. *SIAM Journal of Applied Mathematics*, 55(3):827–863, 1995.

[30] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[31] Y. Chen, S. Thiruvenkadam, H.D. Tagare, F. Huang, D. Wilson, and E.A. Geiser. On the incorporation of shape priors into geometric active contours. In *Workshop on Variational and Level Set Methods in Computer Vision*, pages 145–152, 2001.

[32] Y. Chen, H.D. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K.S. Gopinath, R.W. Briggs, and E.A. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.

[33] L.D. Cohen. On active contour models and balloons. *CVGIP:Image Understanding*, 53(2):211–218, March 1991.

[34] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[35] T. Cootes, C. Beeston, G. Edwards, and C. Taylor. Unified framework for atlas matching using active appearance models. *Int. Conf. Inf. Proc. in Med. Imaging*, pages 322–333, 1999.

[36] M.G. Crandall and P-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277:1–43, 1983.

[37] M.G. Crandall, L.C. Evans, and P-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282:487–502, 1984.

[38] M.G. Crandall, H. Ishii, and P-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27/1:1–67, 1992.

[39] D. Cremers. *Statistical Shape Knowledge in Variational Image Segmentation*. PhD thesis, University of Mannheim, 2002.

[40] D. Cremers. Dynamical statistical shape priors for level set based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, 2006.

[41] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. In N. Paragios, editor, *IEEE 2nd Int. Workshop on Variational, Geometric and Level Set Methods*, pages 169–176, Nice, 2003.

[42] D. Cremers, T. Kohlberger, and C. Schnör. Nonlinear shape statistics in Mumford-Shah based segmentation. In *European Conference on Computer Vision*, volume 2351, pages 93–108, 2002.

[43] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnör. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.

[44] D. Cremers, T. Kohlberger, and C. Schnör. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36:1929–1943, 2003.

[45] D. Cremers, S.J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. *Pattern Recognition*, 3175:36–44, 2004.

[46] D. Cremers, N. Sochen, and C. Schnör. Multiphase dynamic labeling for variational recognition-driven image segmentation. In *European Conf. on Computer Vision*, volume 3024, pages 74–86, 2004.

[47] D. Cremers, C. Guetter, and C. Xu. Nonparametric priors on the space of joint intensity distributions for non-rigid multi-modal image registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1777–1783, June 2006.

[48] D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, September 2006.

[49] D. Cremers, N. Sochen, and C. Schnörr. A multiphase dynamic labeling model for variational recognition-driven image segmentation. *International Journal of Computer Vision*, 66(1):67–81, January 2006.

[50] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, April 2007.

[51] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, NY, USA, 1991.

[52] M. de La Gorce, N. Paragios, and D.J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[53] M.C. Delfour and J.P. Zolésio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. Advances in Design and Control, SIAM.

[54] A.P. Dempster. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[55] R. Deriche and O. Faugeras. Les EDP en traitement des images et vision par ordinateur. *Traitement du Signal*, 13, 1996.

[56] K.G. Derpanis. A review of vision-based hand gestures. Technical report, Centre for Vision Research - York University, Toronto, Canada.

[57] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[58] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *International Conference on Computer Vision*, Nice, France, October 2003.

[59] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis. Learning dynamics for exemplar-based gesture recognition. In *Computer Vision and Patter Recognition*, Madison, Wisconsin, USA, June 2003.

[60] C.L. Epstein and M. Gage. The curve shortening flow. In A. Chorin and Springer Verlag A. Majda, eds., editors, *Wave Motion: Theory, Modelling and Computation*, New York, 1987.

[61] R. Feris, M. Turk, R. Raskar, K. Tan, and G. Ohashi. Exploiting depth discontinuities for vision-based fingerspelling recognition. In *IEEE Workshop on Real-time Vision for Human-Computer Interaction (in conjunction with CVPR'04)*, 2004.

[62] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004.

[63] FSS. Fédération Suisse des Sourds, 2007. http://www.sgb-fss.ch/.

[64] D.M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[65] Z. Ghahramani. Learning dynamic bayesian networks. In *Adaptive Processing of Sequences and Data Structures, in Lecture Notes In Computer Science*, pages 168–197. Springer-Verlag, 1998.

[66] E. De Giorgi. Free discontinuity problems in calculus of variations. *Analyse Mathématique et Applications (Paris 1988)*, 1988.

[67] E. De Giorgi and L. Ambrosio. Un nuovo funzionale del calcolo delle variazioni. *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur.*, 82(2):199–210, 1988.

[68] E. De Giorgi, M. Carriero, and A. Leaci. Existence theorems for a minimum problem with free discontinuity set. *Arch. Rational Mech. Anal.*, 108:195–218, 1989.

[69] P. Goh and E. J. Holden. Dynamic fingerspelling recognition using geometric and motion features. In *IEEE Int. Conf. on Image Processing*, pages 2741–2744, 2006.

[70] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12): 2247–2253, December 2007.

[71] M. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26:285, 1987.

[72] L. Gui, J.P. Thiran, and N. Paragios. A variational framework for the simultaneous segmentation and object behavior classification of image sequences. In *Proc. Scale Space and Variational Methods in Computer Vision, in Lecture Notes in Computer Science*, pages 652–664, Ischia, Italy, 2007.

[73] L. Gui, J.P. Thiran, and N. Paragios. Joint object segmentation and behavior classification in image sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN, USA, 2007.

[74] L. Gui, J.P. Thiran, and N. Paragios. Finger-spelling recognition within a collaborative segmentation/behavior inference framework. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, August 2008.

[75] L. Gui, J.P. Thiran, and N. Paragios. Cooperative object segmentation and behavior inference in image sequences. *International Journal of Computer Vision*, June 2008.

[76] A. Herbulot, S. Jehan-Besson, M. Barlaud, and G. Aubert. Shape gradient for image segmentation using information theory. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 21–24, 2004.

[77] E. J. Holden, G. Lee, and R. Owens. Automatic recognition of colloquial Australian sign language. In *IEEE Workshop on Motion and Video Computing*, 2005.

[78] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[79] T. Iijima. Basic theory of pattern observation. 1959.

[80] S. Jehan-Besson, M. Barlaud, and G. Aubert. Dream2s: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *International Journal of Computer Vision*, 53(1):45–70, 2003.

[81] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[82] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. IEEE Intl. Conf. on Comp. Vis.*, pages 810–815, 1995.

[83] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[84] I. Kokkinos and P. Maragos. An Expectation Maximization approach to the synergy between image segmentation and object categorization. In *ICCV*, pages 617–624, 2005.

[85] S. Kullback. The kullback-leibler distance. *The American Statistician*, 41:340–341, 1987.

[86] J.-O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.

[87] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on SLCV*, 2004.

[88] F. Leitner and P. Cinquin. Dynamic segmentation : Detecting complex topology 3d-object. In *Proceedings of International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 295–296, 1991.

[89] M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 316–323, June 2000.

[90] T. Lindeberg. *Scale-Space Theory in Computer Vision.* The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1994.

[91] R. Lockton and A.W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *British Machine Vision Conference*, 2002.

[92] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[93] Y. Luo, T.D. Wu, and J.N. Hwang. Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks. *Computer Vision and Image Understanding*, 92:196–216, 2003.

[94] J. Maclean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, and J. Tsotsos. Fast hand gesture recognition for real-time teleconferencing applications. In *In International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 133–140, 2001.

[95] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE PAMI*, 17:158–175, 1995.

[96] C. Manresa, J. Varona, R. Mas, and F. J. Perales. Hand tracking and gesture recognition for human-computer interaction. *Electronic Letters on Computer Vision and Image Analysis*, 5(3):96–104, 2005.

[97] A. Mansouri. Region tracking via level set pdes without motion computation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):947–961, 2002.

[98] D. Marr. *Vision.* Freeman Publishers, 1982.

[99] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of Royal Society London*, volume B207, pages 187–217, 1980.

[100] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In IEEE Computer Society Press, editor, *Proceedings of the 5th International Conference on Computer Vision*, pages 840–845, Los Alamitos, California, 1995.

[101] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[102] J. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhauser, 1995.

[103] D. Mumford and J.Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, 42:577–685, 1989.

[104] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22–26, 1985.

[105] H.H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59–74, 1988.

[106] S. Osher and R. Fedkiw. *The Level Set Method and Dynamic Implicit Surfaces*. Springer Verlag, 2002.

[107] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.

[108] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[109] C. Padden and D.C. Gunsauls. How the alphabet came to be used in a sign language. *Sign Language Studies*, 4(1):10–33, 2003.

[110] N. Paragios. *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision*. PhD thesis, School of Computer Engineering, University of Nice, Sophia Antipolis, France, January 2000.

[111] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97:259–282, 2005.

[112] N. Paragios and R. Deriche. Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, pages 249–268, 2002.

[113] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3): 223–247, 2002.

[114] V. Parameswaran and R. Chellappa. View-invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[115] S. Park and J.K. Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia Systems*, 10:164–179, 2004.

[116] E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[117] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[118] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), July 1990.

[119] Natan Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(6):564–569, 1999. ISSN 0162-8828.

[120] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[121] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle filtering for geometric active contours with application to tracking moving and deforming objects. In *Proc. CVPR*, volume 2, pages 2–9, 2005.

[122] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1470–1475, 2007.

[123] N. Robertson and I. Reid. Behaviour understanding in video: a combined method. In *International Conference on Computer Vision*, Beijing, China, October 2005.

[124] F. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[125] A. Rosenfeld and A.C. Kak. *Digital picture processing, Computer Science and Applied Mathematics.* Academic Press, New York, 1982.

[126] M. Rousson and N. Paragios. Shape priors for level set representations. In *European Conference in Computer Vision*, volume 2, pages 78–92, 2002.

[127] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[128] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

[129] D. Sankoff and J. Kruskal. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley Publishing Company, Inc.

[130] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis.* Cambridge University Press, Cambridge, UK, 2001.

[131] K. Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96:100–128, 2004.

[132] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 32–36, 2004.

[133] J.A. Sethian. *An analysis of flame propagation.* PhD Dissertation, Department of Mathematics, University of California, Berkeley.

[134] J.A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101:487–499, 1985.

[135] J.A. Sethian. Level set methods: an act of violence. *American Scientist*, 1996.

[136] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Sciences.* Cambridge University Press, 1999.

[137] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *Computer Vision and Pattern Recognition*, Madison, Wisconsin, USA, June 2003.

[138] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision, second edition.* Brooks and Cole Publishing, 1998.

[139] T. Starner, J. Weaver, and A.Pentland. Real-time American sign language recognition using desk- and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[140] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[141] Demetri Terzopoulos and Richard Szeliski. Tracking with Kalman snakes. *Active vision*, pages 3–20, 1993.

[142] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W.E. Grimson, and A. Willsky. Model-based curve evolution technique for image segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 463–468, 2001.

[143] A. Tsai, A. Yezzi, and A. S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.

[144] K.T. Tseng, W.F. Huang, and C.H. Wu. Vision-based finger guessing game in human machine interaction. In *In Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics*, Kunming, China, December 2006.

[145] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Segmentation, detection, and recognition. In *ICCV*, pages 18–25, 2003.

[146] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999.

[147] R. Urtasun, D.J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition*, New York, USA, June 2006.

[148] D.D. Vecchio, R.M. Murray, and P. Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automata*, 39(12):2085–2098, 2003.

[149] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[150] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[151] Christian Vogler and Dimitris Metaxas. Handshapes and movements: Multiple-channel ASL recognition. In *Gesture Workshop'03, in Lecture Notes in Artificial Intelligence*, pages 247–258, 2004.

[152] R. von der Heydt, E. Peterhans, and G. Baumgartner. Illusory contours and cortical neuron responses. *Science*, 224:1260–1262, 1984.

[153] Chunli Wang, Wen Gao, and Jiyong Ma. A real-time large vocabulary recognition system for Chinese Sign Language. In *Gesture Workshop*, pages 86–95, 2001.

[154] J. Weickert. *Anisotropic diffusion in image processing.* ECMI Series, Teubner-Verlag, 1998.

[155] R. Weiss and M. Boldt. Geometric grouping applied to straight lines. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, USA, 1986.

[156] A.P. Witkin. Scale-space filtering. In *Proceedings of the Eight International Joint Conference on Artificial Intelingence*, pages 1019–1022, Karlsruhe, Germany, 1983.

[157] C. Wren, A. Azarbayejani, T. Darell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[158] Y. Wu and T.S. Huang. Vision-based gesture recognition: A review. In *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, in Lecture Notes in Computer Science*, pages 103–115. Springer-Verlag, 1999.

[159] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 898–903, 1999.

[160] H. Yi, D. Rajan, and L.T. Chia. A new motion histogram to index motion content in video segments. *Pattern Recognition Letters*, 26(9):1221–1231, 2005.

[161] A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *International Conference on Computer Vision*, Beijing, China, October 2005.

[162] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In *Computer Vision and Patter Recognition*, San Diego, California, USA, June 2005.

[163] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1531–1536, 2004.

[164] H. Yu, G.M. Sun, W.X. Song, and X. Li. Human motion recognition based on neural networks. In *International Conference on Communications, Circuits and Systems*, Hong Kong, China, May 2005.

[165] Hong-Kai Zhao, T. Chan, B.Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[166] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

Curriculum Vitae

Name:	**Laura Ioana Gui**
Degrees:	Bachelor of Science in Computer Science, “Politehnica” University Timişoara, Romania
Address:	Signal Processing Laboratory (LTS5) Swiss Federal Institute of Technology (EPFL) CH-1015 Lausanne Switzerland
Contact numbers:	Tel. +41 21 693 46 22 Fax. +41 21 693 76 00 E-mail: laura.gui@epfl.ch gui.laura@gmail.com
Civil status:	Single
Date and place of birth:	May 7th 1980, Romania
Nationality:	Romanian

Education

Since 2003 **PhD student**
Signal Processing Laboratory (LTS5)
Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland
Doctoral School, GPA: 5.6/6

1998–2003 **Bachelor of Science in Computer Science**
“Politehnica” University Timisoara, Romania
High Performance Scholarship, GPA: 10/10

Experience

2005 **Research Intern**
CERTIS Laboratory
École Nationale des Ponts et Chaussées (ENPC), Marne-la-Vallée, France
Developed new tools and performed tests to assess the feasibility of a joint segmentation/behavior recognition framework from image sequences.
The resulted framework was subsequently developed and implemented during PhD work at the EPFL.
Collaboration with Prof. N. Paragios, 1 month

Since 2004 **Teaching and Research Assistant**
Signal Processing Laboratory (LTS5)
Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
Provided assistance for various courses, 6 semesters.
Collaborated with the “Fédération Suisse des Sourds” for the acquisition of a finger-spelling database.

2002 **Software Developer**
Caatoosee, Timisoara, Romania
Designed and implemented a system for monitoring employee-workload on various projects within a company.
Teamwork, 3 months.

Publications

Journal Papers

L. Gui, J.P. Thiran, and N. Paragios.
Cooperative object segmentation and behavior inference in image sequences.
International Journal of Computer Vision, June 2008.

Conference Papers

L. Gui, J.P. Thiran, and N. Paragios.
A variational framework for the simultaneous segmentation and object behavior classification of image sequences.
Proceedings of Scale Space and Variational Methods in Computer Vision.
In Lecture Notes in Computer Science, pages 652–664, Ischia, Italy, 2007.

L. Gui, J.P. Thiran, and N. Paragios.
Joint object segmentation and behavior classification in image sequences.
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, USA, 2007.

L. Gui, J.P. Thiran, and N. Paragios.
Finger-spelling recognition within a collaborative segmentation/behavior inference framework.
Proceedings of the 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, 2008.

Relevant Skills

Areas of expertise:	image processing, image segmentation, machine learning, gesture recognition.
Programming:	C/C++, Matlab, Java.
Human level:	analytical thinking, interpersonal communication, teamwork abilities.

Languages

Fluent:	**English** (Cambridge Certificate in Advanced English, Grade A and ETS GRE) **French** **Romanian** (mother tongue)
Basic knowledge:	**German** **Portuguese** (Brazilian)