Statistical Shape Knowledge in Variational Image Segmentation

Inauguraldissertation
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.-Phys. Daniel Cremers
aus Freiburg i. Br.

Mannheim, 2002

Dekan:	Professor Dr. Herbert Popp, Universität Mannheim
Referent:	Professor Dr. Christoph Schnörr, Universität Mannheim
Korreferent:	Professor Dr.-Ing. Heinrich Niemann, Universität Erlangen-Nürnberg
Tag der mündlichen Prüfung:	24. Juli 2002

Summary

When interpreting an image, a human observer takes into account not only the external input given by the intensity or color information in the image, but also internally represented knowledge. The present work is devoted to modeling such an interaction by combining in a segmentation process low-level image cues and statistically encoded prior knowledge about the shape of expected objects.

To this end, we introduce the *diffusion snake* as a variational method for image segmentation. It is a hybrid model which combines the external energy of the Mumford-Shah functional with the internal energy of the snake. Minimization by gradient descent results in an evolution of an explicitly parametrized contour which aims at maximizing the low-level homogeneity in disjoint regions.

In particular, we present an extension of the Mumford-Shah functional which aims at maximizing the homogeneity with respect to the motion estimated in each region. We named the proposed variational method *motion competition*, because neighboring regions compete for the evolving contour in terms of their motion homogeneity. Minimization of the proposed functional results in an interlaced optimization of the motion estimates in the separate regions and of the location of the motion boundary.

These purely image-based segmentation methods are extended by a shape prior, which statistically encodes a set of training silhouettes. We propose two statistical shape models of different complexity, both of which are automatically generated from a set of binarized training images. The first one is based on the assumption that the training shapes form a Gaussian distribution in the input space, whereas the second one assumes a Gaussian distribution upon a nonlinear mapping to an appropriate feature space. This nonlinear shape prior permits to simultaneously encode in a fully unsupervised manner a fairly complex set of shapes, such as the 2D silhouettes corresponding to several 3D objects. The feature space is modeled implicitly in terms of Mercer kernels. Our approach constitutes an extension of *kernel PCA* to a probabilistic framework.

In order to make the shape prior independent of translation, rotation and scaling of the contour, we propose an intrinsic alignment of the evolving contour with the training set before applying the shape prior. This generates invariance with respect to these tranformations without introducing additional pose parameters which must be determined by optimization.

Gradient descent on a single energy functional maximizes both the low-level homogeneity criterion in each region and the higher-level similarity of the segmenting contour with respect to the training shapes. The resulting knowledge-based segmentation process has a number of favorable properties: The shape prior compensates for ambiguous, missing or misleading low-level information. It permits to segment objects of interest in images (or image sequences in the case of motion segmentation) which are corrupted by noise, clutter or occlusion. In particular, the nonlinear statistical prior encodes fairly different shapes in high detail, and it generalizes to novel views which were not part of the training set.

Acknowledgements

First of all, I would like to express my gratitude to Prof. C. Schnörr for supervising my dissertation and for introducing me to the fields of computer vision and pattern recognition. Both the atmosphere in his group and the wide range of research topics provided an inspiring environment for my work. Secondly, I want to thank Prof. H. Niemann for serving as an external referee. Moreover, I want to thank the committee members of the disputation, namely Prof. U. Brüning, Prof. G. Steidl, PD. J. Hesser and Prof. H.-P. Butzmann.

I want to thank a number of people who contributed to this work in some way or another. In particular, there is my collaborator J. Weickert from whom I learnt a lot about partial differential equations and their numerical implementation. Through his particular sense of humor, many aspects of research became more vivid during many nightly discussions at our institute. Then there are T. Kohlberger and F. Tischhäuser who collaborated with me during their diploma theses, the parts on nonlinear shape statistics and on the multigrid implementation of the diffusion process evolved from joint work. I thank them for a very intense and fruitful cooperation. I enjoyed many discussions with M. Heiler on kernel methods. I want to thank all the other members of our group for providing a wonderful atmosphere, namely C. Schellewald, J. Keuchel, J. Hornegger, S. Weber, A. Bruhn, M. Bergthold, and T. Brox. In particular, I want to thank J. Keuchel and J. Richter who proofread the manuscript and gave many helpful comments which lead to strong improvements of the exposition.

I want to thank a number of researchers for hospitality and stimulating discussions during various visits at other institutes. First of all, this is the group of P. Bouthemy at the INRIA in Rennes who hosted me for two stays of one week and four weeks. The members of his group integrated me very well. In particular I want to thank E. Mémin, T. Corpetti, B. Cernuschi-Frías, C. Hue, F. Cao, S. Paris, E. Arnaud, C. Barillot, I. Corouge and J.-P. Le Cadre for making my stays very memorable. Secondly, there is C. Kervrann and his colleagues A. Trubuil and K. Kiêu from the image analysis and stereology group at the INRA near Paris. I particularly enjoyed many discussions with C. Kervrann during an entire week. I profited immensely from his experience in the fields of image segmentation and statistical shape models. Thirdly, I want to thank the image group at the DIKU in Copenhagen where I participated in two summer schools. During these I got into many interesting discussions with M. Nielsen, J. Sporring, P. Johannson, N. H. Olsen, D. Witzner, O. F. Olsen, K. S. Pedersen, A. B. Lee, A. Pece, Y.-N. Wu, A. Hyvärinen, R. Kimmel and A. Spira. Fourthly, I want to thank J. Denzler for an invitation to the pattern recognition group at the University of Erlangen. I appreciated the hospitality and many fruitful discussions among others with F. Mattern, C. Drexler and F. Deinzer. Fifthly, I want to thank some colleagues from the physics community for invitations to Hannover, Kiel and Essen to present my work. In particular, I enjoyed many discussions with C. Sobiella, O. Lechtenfeld, J.-C. Claussen, H.-G. Schuster, S. Heusler and F. Haake.

Finally, I am grateful to my girlfriend, my brother and my parents for always supporting me in what I was doing.

Nihil est in intellectu quod non antea fuerit in sensu.

(Based on Aristotle, Metaphysics, 350 B.C.)

Contents

Chapter 1

Introduction

1.1 Knowledge-driven Segmentation

The segmentation of images is one of the central problems in the fields of image processing and object recognition. In this work, segmentation refers to the division or partitioning of the image plane $\Omega \subset \mathbb{R}^2$ into a set of disjoint regions[1] $\{R_i \subset \Omega\}_{i=1,\ldots,m}$:

$$\Omega = \bigcup_{i=1}^{m} R_i, \qquad R_i \cap R_j = \emptyset \;\; \forall\, i \neq j.$$

In general, the goal of segmentation is to discriminate which part of the image plane corresponds to an object of interest, and which part corresponds to the background.[2] In this sense, segmentation is closely related to the problem of object recognition. Depending on the cues that distinguish the object of interest from the background, segmentation can be based on edge information, intensity, color, texture, motion or other information. For example, a human figure may be segmented based on the fact that it is darker than its background, whereas human faces may be segmented based on color. A car driving down the street may be segmented because it is moving in a certain direction while the background is static. Or a zebra may be identified because it has a particular stripe pattern distinguishing it from the grass around it — see Figure 1.1.

Figure 1.1: Examples for intensity, motion and texture segmentation.

[1] Morel and Solimini [134] refer to this image partitioning as *strong segmentation*.

[2] The case of segmenting several objects simultaneously will not be covered in this work.

Figure 1.2: Binarized image of a Dalmatian dog in a background of leaves.[3] The dog is located to the right of the center with its back to the viewer, facing left. The human observer combines low-level intensity information and higher-level previously acquired knowledge for segmenting the image.

In all these cases, some information about the object of interest is used. While the human brain tends to automatically select the appropriate cue — for the moving car, the striped zebra etc. — we will assume that for a given machine vision task, a sensible cue is specified beforehand. However, no matter which low-level cue is used for segmentation, one will always find examples where the object of interest is not correctly segmented because the respective assumption underlying the segmentation approach is not fulfilled: The human figure may not be entirely dark or there may be other dark objects in the background. The car motion may be occluded because it is passing behind a static light post. Or the zebra may be in an environment which contains similar grey value patterns. In such cases, the information extracted from the image is not sufficient to define the desired segmentation. The segmentation process is misled by all the information which violates the respective assumption about the low-level image properties characterizing object and background.

Yet in many cases of missing or misleading low-level information, the human brain tends to still perform a correct segmentation of the given image, thereby identifying the object. Figure 1.2 shows an example of a Dalmatian dog in an environment of fallen leaves and grass.[3] Due to coarse-graining and binarization, the dog cannot be distinguished from the background based on the texture only. However, human observers will generally find the correct segmentation after a while. How is that possible? The reason for this is that the human visual system tends to integrate *low-level* and *high-level* information. In the case of the Dalmatian, it combines the low-level texture information of the input image with the high-level notion of what a Dalmatian looks like. This presumption is supported by the experience that people will more easily recognize the object, once they are told what to look for (i.e. a Dalmatian). Moreover, people who have never seen a Dalmatian might not recognize it in the given image.

[3]This image is ascribed to R. C. James.

The goal of the present work is to model such an interaction between the low-level information contained in an external input image and the high-level internal information about the object of interest, which is acquired beforehand and statistically represented during a learning process.

In the human visual pathway, the integration of low-level cues and internally represented high-level information arises through the neural activity in several strongly interconnected layers of neurons, starting at the retina, over the lateral geniculate nucleus to various layers of the visual cortex with strong feedback connections at each level.

Rather than emulating the neuronal architecture of the human visual pathway and simulating the corresponding highly nonlinear dynamics, we decided for a mathematically simpler fusion of external and internal information in a *variational framework*. The reason for this choice is twofold: Firstly, we believe that the computational overhead introduced by modeling individual neurons would impede the treatment of higher-level concepts, such as statistical models of shape. And secondly, for the purpose of improving machine vision systems, it may be sufficient to adopt certain general concepts from the human visual system — in our case the *fusion of external and internal information*.

In this work, we make the following assumptions in order to focus on a more specific case of combining external and internal information:

- For simplicity, we will restrict the problem of segmentation to that of finding a single closed curve $C : [0, 1] \to \Omega$ which segments the image plane Ω. However, extensions to several curves and several objects are conceivable.

- The higher-level internal knowledge will only comprise the *shape of the segmenting contour*. This certainly limits the applicability of our approach, since many objects such as faces are not primarily defined by their silhouette. However, we are currently investigating in how far the internal knowledge can be extended to also encompass region information.

- As external input we will only consider planar grey value images or image sequences:

$$f : \Omega \longrightarrow \mathbb{R}_+, \quad \text{or} \quad f : \Omega \times \mathbb{N}_+ \longrightarrow \mathbb{R}_+.$$

  However, all results are easily extended to color and other multi-spectral images. Extensions to 3D images are also conceivable, yet they are not straight forward since matters such as shape alignment and similarity invariance of the shape information are more complicated in higher dimension.

In the remaining parts of this chapter, we will make some more general remarks on the relation between *variational methods* and the paradigm of *Bayesian inference*, about the pros and cons of *explicit* versus *implicit* contour representations, and about different notions of *shape dissimilarity*. These should help to characterize our approach before we specify the contributions of our work in more detail.

1.2 Variational Methods and Bayesian Inference

In this work, we segment a given image or image sequence f by finding contours C which minimize functionals of the form

$$\boxed{E(C) = E_{image}(f, C) + \alpha\, E_{knowledge}(C).} \tag{1.1}$$

This cost functional or energy is made up of two components: The first one measures how well the contour segments a given input image, based on the external grey value information given by the image f and a particular segmentation cue such as homogeneous grey value or motion information. The second term represents the higher-level knowledge about the object of interest which was previously acquired in a learning process — e.g. in the case of the Dalmatian in Figure 1.2, it would ideally measure how different the segmenting contour C is from a Dalmatian. The parameter $\alpha \geq 0$ permits to define the weight between external and internal information. For $\alpha = 0$ the system only takes into account the external input, whereas for $\alpha > 0$ the internally represented knowledge will influence the segmentation process.

Many approaches to image segmentation are modeled in a probabilistic framework. For completeness, we want to point out that the variational approach (1.1) is equivalent to the approach of Bayesian inference: Given an input image (or image sequence) f, one maximizes the *posterior probability*

$$P(C|f) \;=\; \frac{P(f|C)\; P(C)}{P(f)}. \tag{1.2}$$

Maximizing this conditional probability with respect to the contour C for a fixed input f is equivalent to minimizing its negative logarithm:

$$-\log P(C|f) \;=\; -\log\big(P(f|C)\big) \;-\; \log\big(P(C)\big) \;+\; \text{const.}$$

The equivalence to the variational approach (1.1) is obtained by identifying $E_{image}(f, C) = -\log\big(P(f|C)\big)$ and $\alpha\, E_{knowledge}(C) = -\log\big(P(C)\big)$.

The above equivalence shows how the external energy is related to the probability of a grey value distribution f given a contour C. Moreover, the internal energy $E_{knowledge}$ can be interpreted as the negative log-likelihood of the *a priori probability* for a given contour C. As we will see further on, this can be a rather general prior which simply states that longer contours are less probable, like

$$P(C) \;\propto\; e^{-\alpha|C|},$$

where $|C|$ is a measure of the contour length. But it can also be a more elaborate *shape dissimilarity measure*

$$P\big(C \,|\, \{C_i\}\big),$$

which is constructed from a set of training silhouettes $\{C_i\}_{i=1,\ldots,m}$.

Independently of the chosen paradigm — the variational formulation (1.1) or the maximum a posteriori (MAP) formulation (1.2) — different methods can

be employed to obtain the extrema. There exists a number of *global* optimization techniques such as simulated annealing [81], mean field annealing [79] and graduated nonconvexity [18]. In this work, however, we will only consider *local* optimization techniques. There are several reasons for this choice:

- In numerical studies, we found that the functionals we study tend to have few minima, such that global optimizers can be expected to produce similar results.
- Compared to many other optimization problems, in our case even local extrema generally correspond to sensible segmentations of a given input image. In fact, a local optimum is often more desirable, since it corresponds to the "closest" segmentation for a given initialization. For example, if there are several objects in an image, they may be obtained sequentially by local optimization with different initial contours.
- Since our goal is to model the interaction of external and internal information in a segmentation process, we avoided the additional complications of global optimizers: Firstly, most standard implementations cannot guarantee to find the global optimum, especially in the case of high dimensions.[4] And secondly, tuning the parameters needed by most global optimization methods can be tedious.
- Especially in high dimensions, global optimization tends to be much slower than a local scheme. We found that by using local optimization schemes, we are able to obtain performances close to real-time for many applications. Although this is not our main goal, it tends to facilitate experimentation and makes online demonstrations feasible.

1.3 Implicit versus Explicit Contours

In this work, we decided for an explicit representation of the contour C in (1.1). This choice shall be briefly justified in the following.

For the representation and temporal evolution of contours, one can choose between explicit and implicit representations. Implicit contours C are contours of the form

$$C = \{x \in \Omega \mid \phi(x) = 0\}. \tag{1.3}$$

This means that the contour C is given by the zero level set of a function $\phi : \Omega \to \mathbb{R}$. In the case of algebraic implicit contours the function ϕ is given by a polynomial [75, 147]. An alternative is to approximate arbitrary functions ϕ numerically on a grid. This approach has become quite popular with the introduction of level set methods [143], which permit the numerical propagation of surfaces ϕ with a curvature-dependent speed.

A number of well known segmentation methods have been (re)formulated in terms of implicit contours. The initial contour is embedded in a surface, for example by the signed distance function. The contour evolution is replaced

[4] In our case, optimization is usually done in more than 200 dimensions.

by an evolution of the embedding surface, and the corresponding contour at a given time is obtained by determining the zero level set of the evolving surface (cf. [32, 108, 35, 195]).

The alternative to implicit contour representations are explicit ones. These can for example be implemented by a set of discrete marker points [200] which are then evolved over time. In the computer vision community the most popular explicit contour representation for shape modeling, segmentation and tracking are spline contours [129, 39, 96, 71] of the form

$$C : [0,1] \longrightarrow \Omega, \qquad C(s) = \sum_{i=1}^{N} p_i \, B_i(s),$$

where $p_i \in \mathbb{R}^2$ are the control points and $B_i(s)$ are appropriate spline basis functions of some fixed degree. Linear, quadratic or cubic spline basis functions are most commonly used.
Explicit contour representations have several advantages and disadvantages:

+ The *computational cost* of evolving a parametric contour is much lower than that of evolving the associated surface, because it amounts to updating a fairly small number of parameters rather than the full two-dimensional embedding function (or at least a narrow band of this function around each contour).
+ The explicit nature of the contour provides a *compact representation of a given shape*. This permits to directly perform shape analysis, shape alignment and the statistical modeling of a distribution of training shapes (cf. [83, 67, 47]).
− In the propagation of an *explicit* contour, *numerical instabilities* can arise if control (or marker) points move too close together. Generally one needs to revert to a regridding mechanism [200] or introduce some additional force which prevents the clustering of control points (cf. [56]).
− During the evolution of the embedding surface ϕ, an *implicit* contour can undergo *topological changes* such as splitting and merging, which do not need to be modeled explicitly, because topological changes of the contour C do not imply topological changes of the embedding function ϕ.

  However, in many applications of knowledge-driven segmentation, topological changes of the contour can be excluded *a priori*. Usually a given prior shape information restricts the evolving contour to a manifold of familiar shapes, in which no contour splitting or merging can occur. Therefore the constraint imposed by the shape prior can be considered much stronger than that imposed by the constancy of the contour topology.

In this work, we decided for an explicit contour representation because it facilitates the modeling of statistical shape priors.

Recently, there have been some efforts to model shape statistics on the basis of implicit contour representations [120]. However, there the training shapes are embedded by the signed distance function and the distribution of embedding surfaces is modeled statistically. Apart from the fact that this drastically

increases the dimension of the input space, it is unclear in what way the surface representation affects the shape learning, since only the zero level set of the surface corresponds to a *perceivable* shape. Matters such as alignment and reparameterization are not satisfactorily solved. Moreover, although the segmentation process with an implicit contour permits a contour splitting, the separate contours cannot be treated as statistically independent. This means that the proposed shape influence cannot be used for the segmentation of several (independent) objects in a given input image.

On the other hand, there have been several approaches to model topological changes for explicit contours [119, 127, 115, 64]. These methods are necessarily heuristic and additional decision parameters have to be introduced, too. Yet they have been demonstrated to work well in many practical applications.

1.4 Two Distinct Notions of Shape Dissimilarity

The goal of this work is to integrate appropriate statistical priors on the shape of the segmenting contour into the segmentation process. To this end, we will represent a set of training shapes in a vector space and derive different shape dissimilarity measures on the basis of the distribution of the training shapes.

For clarity, we distinguish two very different notions of *shape dissimilarity*:

1. *The dissimilarity of two different shapes (or contours)*: Such a measure can incorporate low-level geometric information in terms of the *deformation energy* needed to bend or stretch one contour into the other, as for example proposed in the work of Basri et al. [9]. It can also rely on higher-level concepts such as the correspondence of subparts — for example the relative position of corresponding legs may be different from one human figure to another. The resulting dissimilarity measure can incorporate operations known from syntactic or string matching such as substitution, deletion or insertion. This has been done among others by Gdalyahu and Weinshall [78]. Moreover, cognitive psychophysical concepts such as the correspondence of maximal convex or concave subparts can be integrated in such distance measures, as proposed by Latecki and Lakämper [116].

2. *The dissimilarity of a given shape with respect to a set of training shapes*: If the first notion of distance between two shapes can be formulated as a metric induced by a scalar product, then the training shapes are part of a Hilbert space. This permits to estimate a shape probability distribution underlying this set of training shapes. The associated energy density, given by the negative logarithm of this probability density, can be interpreted as a shape dissimilarity measure.

The second notion of dissimilarity is obviously based on a choice for the first one. Yet, these two notions are complementary: The first one assumes some knowledge about how one shape is deformed into another — this becomes apparent once parameters have to be specified to determine the cost of e.g. deletion, insertion or bending. In contrast, the second notion of dissimilarity is a statistical one which is induced by a set of example shapes.

In this work, we focus on the second notion of shape dissimilarity, because it is closer to the paradigm of *learning from examples*. For computational efficiency, we will revert to very simple measures of the distance between two shapes. We will present two measures of *statistical* shape dissimilarity which differ in their complexity. Numerical experiments will show that these permit to encode fairly complex and detailed shape information if a sufficient number of training shapes is given.

1.5 Related Work

There already exists a vast amount of literature on many image segmentation methods. To survey this entire field is beyond the scope of this work. A brief review of *variational* approaches to segmentation will be given at the beginning of Chapter 2, with a particular focus on the ones our own approach is based upon. Similarly, references to related work in the field of motion segmentation are postponed to Chapter 5.

The principles underlying vision in biological systems have been studied by neurobiologists and psychophysicists, one of the earliest of which was Helmholtz [92]. The idea to treat biological vision and computer vision as a joint problem was propagated among others by Marr [123]. The interpretation of computer vision as a problem of Bayesian inference was pioneered in particular by a group of researchers at Brown University, namely Grenander, Geman, Mumford and co-workers (cf. [87, 81, 203]).

The study of shape has a long tradition. An early writing on shape is that of Galilei [77], who compared bones of differently sized animals, finding that for stability reasons they differ not only in size but also in their shape. An early work dealing with the dissimilarity of two shapes is that of D'Arcy Thompson [173], who showed that one species of fish (the Diodon) could be geometrically transformed into another (the Orthagoriscus). Similarly, he *warped* the skull of a human into that of a chimpanzee or a baboon by deforming an underlying Cartesian grid. This technique has been refined with deformations in terms of thin-plate splines by Bookstein in [22]. Many of the key ideas underlying statistical shape analysis were developed by Kendall [103] and Bookstein [21]. For a detailed review we refer to [67].

Statistical models of shape variation for computer vision were pioneered by Grenander [88]. Shape approximation by spline curves was propagated among others by Menet et al. [129] and Cipolla and Blake [39]. Increasingly more elaborate models of shape and appearance have been proposed by Cootes, Taylor and co-workers under the names of *point distribution model*, *active shape* and *active appearance model* [44]. Baumberg and Hogg [10] presented a method for automatic shape acquisition using background subtraction and a spline-based shape analysis.

More recently, a number of *nonlinear* models of shape variation were presented, i.e. models where the permissible shape variation is not constructed by a linear combination of eigenmodes. Among these are mixture models by Cootes et al. [48] and the related hierarchical point distribution models by Heap

and Hogg [91], hybrid models using both Cartesian and polar coordinates [90], nonlinear extensions by Sozou et al. using multi-layer perceptrons [169] or polynomial regression [168], and kernel principal component analysis by Romdhani et al. [150] and Twining and Taylor [179].

Applications of shape models in segmentation or matching were proposed among others by Yuille [197], Yuille and Hallinan [198], Grenander [88], Staib and Duncan [170], Cootes et al. [46], Kervrann et al. [105, 106], Wang and Staib [183], Duta et al. [70], and Leventon et al. [120]. We will not go into detail about the contributions of each of these works. For a more detailed review, we refer to [17].

In comparison to the above approaches, the main points of our work are:

- Statistical shape models are usually incorporated in edge-based segmentation methods, whereas we use region-based methods.[5] In Chapter 2, we will discuss differences between edge-based and region-based variational segmentation methods and introduce our own segmentation approach, the *diffusion snakes*, on the basis of the Mumford-Shah functional [136].
- The region-based segmentation method can be easily extended to different low-level segmentation cues such as texture, color or motion. In Chapter 5, we will demonstrate this by introducing statistical shape information into a novel framework for variational *motion* segmentation.
- We do not restrict the segmenting contour to the low-dimensional subspace of a few deformation modes (cf. [198, 16]). Although such a compact representation tends to reduce the computational effort, it has certain disadvantages. Firstly, the effect of the prior cannot be continuously decreased. Secondly, a shape probability which is non-vanishing only in a low-dimensional subspace is less faithful from a probabilistic point of view — see Section 3.3. And finally, extensions to more general (nonlinear) probabilistic shape models are not straight-forward, because in these cases, finding a low-dimensional parametric description of permissible shape variations (such as the principal eigenmodes in the linear case) may be entirely infeasible.
- Nonlinear models of shape variation have appeared only fairly recently. Therefore, to our knowledge, there has not been any work of incorporating nonlinear shape statistics into a region-based segmentation method. This will be presented in Chapter 4.

1.6 Contributions

Different parts of the work presented here have been published on various occasions [57, 58, 56, 50, 51, 52, 59, 54, 53, 55]. Some of this work resulted from cooperations with two diploma students, namely Timo Kohlberger and Florian Tischhäuser. Therefore some results concerning nonlinear shape statistics have

[5] Note also, that in many applications of shape models in computer vision, a fitting of open contours to image structures is performed. This does not produce a (strong) segmentation in the sense of a partitioning of the image.

appeared in [112], and results on multigrid implementations of the diffusion process in [177].

The main contributions can be split into four components which are contained in the Chapters 2 through 5. For better readability these chapters are mostly self-contained.

Diffusion Snakes

In Chapter 2, we present a variational method for image segmentation which can be considered a hybrid of two models: The functional combines the external image energy of the Mumford-Shah functional [136] with the internal energy of the classical *snake* [102]. Due to the underlying diffusion process, we named it *diffusion snake*. The corresponding snake-like implementation of the piecewise constant Mumford-Shah model is called the *simplified diffusion snake*.

In numerous experimental results, we show that these region-based snakes are fundamentally different from edge-based approaches such as the classical snake: The issues of image smoothing and optimal edge placement are separated in the variational formulation, such that noise robustness and large basins of attraction are obtained without destruction of relevant image information such as the precise location of edges and corners. Moreover, during minimization by gradient descent the contour converges over fairly large spatial distances although there are no balloon-terms [40] in the functional which would induce a bias towards expansion or contraction. On the contrary, we demonstrate that for the same parameter value the contour can both expand and contract depending on the image information. We also compare segmentation results obtained by the diffusion snake and the simplified diffusion snake with those obtained by a level set implementation of geodesic active contours [32, 108].

Diffusion Snakes with Linear Statistical Shape Prior

In Chapter 3, we propose to extend the diffusion snake functional by a *statistical shape energy* which favors the formation of familiar contours. Familiarity is defined on the basis of a set of binarized training shapes. We discuss the issues of automatic contour extraction, alignment and shape learning. We assume that the set of training shapes are distributed according to a Gaussian probability density. In contrast to most active shape models, we do not restrict the contour deformation to a low-dimensional subspace of the first few eigenmodes. Due to a regularization of the sample covariance matrix we obtain a finite non-zero probability in the full space of possible contour deformations. The covariance regularization is related to *probabilistic principal component analysis* or *sensible PCA* [131, 155, 176]. However, we propose a choice of the regularizing constant which deviates from that proposed in [131, 176].

We present numerous ways to incorporate in the variational approach an invariance of the shape prior with respect to certain transformations of the contour. In particular we discuss a framework of *learning invariances*, which conforms with the paradigm of *learning from examples*. We show that robustness to some transformations can be learnt, but that this method cannot be

extended to full similarity invariance. As a remedy, we propose a closed-form solution for incorporating similarity invariance into the variational approach. It is formulated on the basis of the spline representation of the contour and has the advantage that *no additional parameters* must be introduced to account for translation, rotation or scaling. We compare this method of incorporating invariance to alternative approaches known from the literature.

Experimental results demonstrate the influence of the linear shape prior on the segmentation process. We show examples where the application of different shape priors permits to parse an object into its constituent components. Similarity invariance of the shape prior is demonstrated. Moreover, we show how the shape prior permits to cope with noise, clutter and occlusion.

Shape Statistics in Feature Space for Segmentation

In Chapter 4, we present a nonlinear generalization of the shape dissimilarity measure which is based on the assumption that the training shapes are distributed according to a Gaussian probability density after a nonlinear mapping to an appropriate feature space. The mapping to the feature space is modeled implicitly in terms of Mercer kernels [130, 49]. The proposed dissimilarity measure can be interpreted as an extension of kernel PCA [164] to a probabilistic framework.

Compared to alternative nonlinear shape models, the proposed method does not assume any prior knowledge about the type of nonlinearity. Moreover, *no prior clustering or classification* of the training shapes is necessary. The model contains a single free parameter for which automatic estimates are given. Combined with the external image energy in the diffusion snake functional, this shape energy restricts the contour evolution to a submanifold of familiar shapes.

Experimental results show that the *nonlinear shape prior* is far more powerful than the linear one since it permits to encode a large variety of different shapes, such as those corresponding to different objects and different views of a 3D object. The capacity of the nonlinear shape prior to cope for noise, clutter and occlusion of the objects of interest is demonstrated in several artificial and real-world applications. The restriction of the contour to the learnt manifold during applications in segmentation and tracking is demonstrated by appropriate projections of both the training shapes and the evolving contour. In this way we are able to verify the *statistical nature* of the nonlinear shape prior, namely that it can *encode in high detail a large set of fairly different training silhouettes* while still permitting a *generalization to novel views* which were not part of the training set.

Some relations of the proposed feature space distance to classical methods of density estimation are discussed in Appendix C.

Motion Competition

In Chapter 5, we propose an extension of the Mumford-Shah functional to the problem of segmenting an image sequence into regions of *piecewise homogeneous motion*. Again we present an implementation with an explicit contour similar

to that of the diffusion snakes. This permits an incorporation of a statistical prior on the shape of the motion discontinuity curve. We focus on the two cases of piecewise constant and piecewise affine motion, however other linear parametric models could be used as well.

In experimental results we demonstrate the fundamental differences between motion and grey value segmentation. We compare segmentation results obtained with the models of piecewise constant grey value, piecewise constant motion and piecewise affine motion. We experimentally verify the properties of the proposed motion segmentation: During minimization, the motion discontinuity curve converges over fairly large distances, and the motion estimates are updated in alternation so as to gradually separate the different motion fields. In particular, the method permits to segment two differently moving regions, as given in the case of moving objects captured by a differently moving camera.

As in the case of grey value segmentation, we demonstrate the capability of the shape prior to cope with incomplete motion information due to noise and compensate for the (more fundamental) limitations induced by the aperture problem. We demonstrate that due to the statistical shape prior, an object of interest can be segmented on the basis of its relative motion although the motion information is partially occluded.

Conclusion and Appendix

In Chapter 6, we briefly review the results of the present work. We discuss a number of limitations of the proposed methods and point out directions of ongoing and future work.

In order to not break the flow of the argument, certain topics were postponed to the Appendix. Part A contains some remarks on spline distance approximations. Essentially we justify the use of the Euclidean distance between spline control point polygons as an approximation of a more elaborate spline distance. Part B contains details on a multigrid implementation of the inhomogeneous diffusion process underlying the contour evolution of the diffusion snake. We define appropriate restriction and prolongation operators to model the transfer between coarse and fine grids and present the stencils used in the numerical implementation. Part C contains some remarks on feature space distances and their relation to classical methods of density estimation. In particular, we show that the Euclidean distance associated with a spherical (isotropic) Gaussian distribution in feature space corresponds to a Parzen estimate in the original space. We then present some preliminary insights characterizing the Mahalanobis distance associated with an ellipsoidal (anisotropic) Gaussian distribution in feature space.

Chapter 2

Variational Image Segmentation

A large variety of approaches have been proposed to tackle the problem of image segmentation. In the following, we will briefly review some of the *variational methods*. Explicit variational formulations have a number of advantages (cf. [134]):

- The variational approach presents explicitly the quantity which is optimized. In contrast, many heuristic approaches propose an application of successive image processing steps or a combination of different tools. However, in order to modify or improve a given segmentation method, one should know what precisely is optimized.
- Most segmentation methods *can* be formulated in terms of an explicit functional which is minimized.
- The variational approach automatically offers a quantitative criterion for comparing the quality of two given segmentations in a self-consistent way.
- The variational formulation can be deduced from a classical axiomatization of image processing given by *multiscale analysis* [4].
- Many of the results presented in this work will show that the variational framework is well suited to model the fusion of external image information and internally represented prior knowledge in a single segmentation process. As discussed in Section 1.2, this variational integration of external and internal information is equivalent to the probabilistic framework of Bayesian inference.

2.1 From Edges to Multiscale Image Analysis

Some of the earliest approaches to image segmentation are based on the low-level feature of *edges* [25, 125, 132], where edges are commonly defined as regions where the magnitude of the image gradient is maximal or where the Laplacian

of the image shows zero-crossings. They indicate locations of intensity discontinuities which are assumed to correspond to discontinuities in the geometry.

A fundamental property of edges is that they are only defined with respect to the *spatial scale* on which the intensity discontinuity takes place. Moreover, the detection of edges by differentiation of the intensity function is very sensitive to noise. To address these two difficulties, one commonly reverts to multiscale filtering and multiscale edge detection. Essentially this means that the image is first smoothed at various scales and the edges are determined afterwards in terms of the maxima of the gradient or the zero-crossings of the Laplacian. Equivalently one can directly convolve the input image with suitable derivatives of Gaussian-like filters. The width of the filter determines the spatial scale on which edges are to be detected. The family of images obtained by filtering the input image at various scales induces the notion of *scale space*.

The theory of linear multiscale filtering has been extensively studied [153, 124, 192, 110, 199, 30]. An early axiomatic derivation of linear Gaussian scale-space was given by Iijima in 1962 [99, 100], see also [187]. As pointed out by Koenderink [110], linear Gaussian smoothing of an image f at various scales is equivalent to solving the heat equation

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \triangle u(x,t) \\ u(x,0) &= f(x) \end{aligned} \tag{2.1}$$

with the input image f as the initial condition, as the solution to (2.1) is given by $u(x,t) = g_t \star f$, where $\star$ denotes the convolution and $g_t(x) = \frac{1}{4\pi t} \exp\left(-\frac{||x||^2}{4t}\right)$.

In order to introduce a non-trivial steady state into equation (2.1), it can be extended by an inhomogeneity:

$$\left(\frac{\partial}{\partial t} - \lambda^2 \triangle\right) u = f - u, \tag{2.2}$$

with the scale parameter $\lambda \geq 0$. This is the gradient descent evolution for the functional

$$E(u) = \int_\Omega (f-u)^2 \, dx \; + \; \lambda^2 \int_\Omega |\nabla u|^2 \, dx. \tag{2.3}$$

For a given scale parameter λ, the minimum of (2.3) corresponds to a smoothing of the input image f at the given scale. The functional (2.3) can be considered the most simple example of an entire class of variational formulations for image processing problems, which consist of two terms: The first one is an *approximation* or *fidelity* term which assures that the minimum is in some sense similar to the input image and the second term is a *regularity* or *smoothness* term, which guarantees that the minimum is as smooth or regular as possible (cf. [174, 172]).

The linear diffusion equation (2.2) aims at smoothing all image structure on the spatial scale λ. In practice, however, one would like to smooth noise without loosing the information about the location of edges and other relevant image features. In this case one can revert to adaptive or nonlinear filtering

[145, 139, 159, 138, 34, 184, 85, 142]. A corresponding variational approach is given by functionals of the form

$$E(u) = \int_\Omega (f-u)^2\,dx \;+\; \lambda^2 \int_\Omega G\left(|\nabla u|^2\right) dx, \tag{2.4}$$

where the function G is generally some kind of robust estimator of the edge strength.[1] Functional (2.4) is commonly referred to as the Perona-Malik model [145]. For $G(s) = s$, (2.4) reduces to the linear case (2.3). In contrast to the linear model, edges or more generally areas of large image gradient tend to be preserved, if the function G rises more slowly than the linear function. In particular, if $G(s) = \sqrt{s}$, the smoothness term is called the *total variation* (cf. [142]). The gradient descent evolution associated with the functional (2.4) is a nonlinear diffusion equation, with a diffusivity which depends on the magnitude of the image gradient.

This approach can be extended even further to models of nonlinear *anisotropic* diffusion with a matrix-valued diffusivity D which also takes into account the *direction* of the image gradient (cf. [184]). The resulting diffusion process smoothes the image in direction of the level lines, thereby enhancing the edges. However, variational formulations underlying such anisotropic diffusion processes only exist for the case of vector-valued images [186].

2.2 Edge-based Segmentation Approaches

2.2.1 Snakes

The variational approaches (2.3) and (2.4) both produce simplified versions of a given input image f, in the sense that the input image is smoothed at the spatial scale λ. In the case of (2.4), smoothing adapts to the local image gradient. However, neither of these approaches produces a *strong segmentation* of the input image, as no partitioning of the image plane into disjoint regions is performed. Even if edges are detected in an image with the help of multiscale filtering: How should they be linked in order to obtain a segmentation of the image? This question has been addressed by a number of researchers [152, 189].

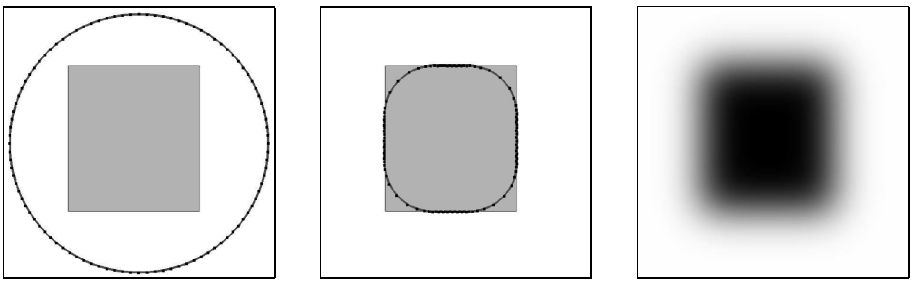
A variational approach to solve the problem of edge linking is the classical *snake* functional proposed by Kass, Witkin and Terzopoulos [102]:

$$E(C) = \int \left\{ \frac{\nu_1}{2} |C_s|^2 + \frac{\nu_2}{2} |C_{ss}|^2 - |\nabla f(C)|^2 \right\} ds. \tag{2.5}$$

Here $C(s)$ denotes an explicit parametric closed curve, and C_s and C_{ss} denote the first and second derivative with respect to the curve parameter. The first two terms in (2.5) can be interpreted as an *internal energy* of the contour, measuring the length of the contour and its stiffness or rigidity[2]. Both are weighted

[1] In [34] it is shown that a more consistent method is obtained (for the model without the fidelity term) if the function G is applied to the absolute value of the Gaussian-presmoothed image $u_\sigma = g_\sigma \star u$ (where g_σ is a Gaussian of width σ).

[2] From a survey of a number of related publications and from our personal experience, it appears that the rigidity term is not particularly important, such that one commonly sets $\nu_2 = 0$.



Initial contour | Final segmentation | Gaussian smoothed input image

Figure 2.1: Initial and final contour for the snake (2.5). The input image is a black box on white ground (depicted in grey for better visibility). In order to create a sufficiently large basin of attraction, the input image was Gaussian-smoothed as shown on the right. Due to this smoothing, the edge gradient is noticeable at a longer range. However, the smoothing also blurs details such as the corners. This dilemma arises since the original snake is only defined on a single scale.

with nonnegative parameters ν_1 and ν_2. The last term is the external energy which accounts for the image information, in the sense that the minimizing contour will favor locations of large image gradient. Minimization of (2.5) by gradient descent results in the evolution equation[3]

$$\frac{dC(s,t)}{dt} = -\frac{dE}{dC} = \nu_1 C_{ss} - \nu_2 C_{ssss} + \nabla|\nabla f(C)|^2. \tag{2.6}$$

The last term in this evolution equation drives the contour to areas of high image gradient. As discussed in Section 2.1, edges are always defined on a certain spatial scale. This is precisely the weakness of the snakes: Depending on the initialization, the input image f needs to be appropriately presmoothed in order to create a sufficiently large basin of attraction for the snake to converge. However, as discussed in Section 2.1, linear presmoothing also destroys the exact location of edges, so that the final segmentation tends to "blur" the object of interest by smoothing sharp corners and small details — see Figure 2.1.

Although the snake solves the problem of edge linking, it is not defined in a multi-scale framework. A remedy which we found to work in practice is to run the snake evolution on input images f_σ which are smoothed on several scales $\sigma_1 > \sigma_2 > \sigma_3 > \cdots > \sigma_n$. Moreover, additional terms can be introduced to draw the contour towards corners [17]. However, these remedies are not very elegant from a theoretical point of view. In addition, Gaussian smoothing at several scales is a rather time consuming process, while speed is one of the strengths of the explicit snakes.

[3] In a number of publications, including [102], the signs of the individual terms in the evolution equation (2.6) are partially incorrect.

2.2.2 Balloons

Another way to drive the contour in (2.5) towards the desired segmentation over larger distances is to introduce an additional force which can either shrink or expand the contour. These so-called *balloons* proposed by Cohen and Cohen [40] are obtained by adding an extra term to the functional (2.5) which favors regions of a certain size:

$$E(C) = \int \left\{ \frac{\nu_1}{2} |C_s|^2 + \frac{\nu_2}{2} |C_{ss}|^2 - |\nabla f(C)|^2 \right\} ds \; + \; \nu_3 \int\limits_{\Omega_i} dx, \tag{2.7}$$

where Ω_i is the region inside the contour. Depending on the sign of the parameter $\nu_3 \in \mathbb{R}$, this induces an additional driving force along the contour norma which either shrinks ($\nu_3 > 0$) or expands ($\nu_3 < 0$) the contour.

Obviously, this additional term reduces the generality of the snake. In a practical application, one needs to *know* whether the initial contour is located inside the object of interest or whether it encloses the object. Moreover, depending on the magnitude of ν_3, the final segmentation will show a bias towards the inside or the outside of the segmented object. In practice, this bias can be minimized by decreasing the magnitude of ν_3 during the evolution.

2.2.3 Geodesic Snakes

The classical snake (2.5) is formulated on the basis of an explicit contour. As discussed in Section 1.3, this has several disadvantages, the main one being the topological rigidity, i.e. the fact that no contour splitting or merging is possible. Rather than explicitly incorporating mechanisms which permit topological changes of the explicit contour [119, 127, 115, 64], one can embed the contour evolution into an evolution of a surface, where the contour is given by the zero level set of the respective surface. These so-called *geodesic snakes* have been simultaneously proposed by Caselles, Kimmel and Sapiro [32, 33] and by Kichenassamy, Kumar, Olver, Tannenbaum and Yezzi [108].

In [33] the classical snake (2.5) is first generalized to a larger class of edge detectors by replacing $-|\nabla f|^2$ with $g\left(|\nabla f|\right)^2$, where $g : \mathbb{R} \to \mathbb{R}$ is a strictly decreasing function which asymptotically vanishes: $\lim\limits_{s\to\infty} g(s) = 0$. Under some additional assumptions (in particular $\nu_2 = 0$) it is then shown that by *Maupertuis' principle of least action*, the minimization of the snake energy amounts to finding paths of minimal "weighted" distance

$$\min_C \int\limits_0^1 g\left(|\nabla f(C(q))|\right) \, |C_q(q)| \, dq, \tag{2.8}$$

where the infinitesimal contour length $dC = |C_q(q)| \, dq$ is weighted by the inverse edge strength[4] $g\left(|\nabla f(C(q))|\right)$. The minimization problem (2.8) can be interpreted as finding a geodesic curve (i.e. a curve of smallest length) in a Riemannian space, the metric tensor of which is induced by the input image f.

[4] We speak of *inverse* edge strength, because g decreases with increasing edge strength.

Gradient descent on (2.8) results in the curve evolution equation

$$\frac{\partial C(t)}{\partial t} = g\left(|\nabla f(C)|\right)\kappa\,\boldsymbol{n} - (\boldsymbol{n}\,\nabla g)\boldsymbol{n}, \tag{2.9}$$

where $\boldsymbol{n}$ denotes the unit inward normal on the contour and κ its Euclidean curvature. The first term represents a *Euclidean curve shortening flow* which is weighted by the inverse edge strength g. The second term is the normal component of the force towards areas of large image gradient. In practice, the first term is extended by replacing κ with $\kappa + c$, where c is an appropriate constant. This induces a similar expansion or shrinking force along the normal as was obtained for the balloon model (2.7). Here again, it appears that a choice of the sign of c implies prior knowledge on whether the curve is to shrink towards an object which is initially enclosed (inward flow) or rather to expand towards an object which initially encompasses the contour (outward flow).[5]

The evolution equation (2.9) can be embedded into an image evolution of the form[6]

$$\frac{\partial \phi(x,t)}{\partial t} = |\nabla\phi|\,\mathrm{div}\left(g(|\nabla f|)\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{2.10}$$

which implies that all level sets of the function $\phi(x,t)$ evolve according to equation (2.9). The contour of interest is usually encoded as the zero level set of ϕ — see equation (1.3). The advantage of this implicit formulation is that the contour C can undergo topological changes which do not need to be modeled explicitly. This permits the segmentation of several objects in a given image. A modification of (2.10) where $g(|\nabla f|)$ is replaced by $g(|\nabla u|)$ was proposed under the name of *self-snake* in [156].

2.3 Region-Based Segmentation Approaches

2.3.1 The Chicken and Egg Dilemma

The above approaches can be considered *edge-based* approaches in the sense that the contour is essentially drawn to the nearest maxima of the input image gradient. As discussed in Section 2.1, the input image is generally presmoothed at a scale σ to obtain more reliable edge information — this creates larger basins of attraction and a certain noise robustness since edges at scales smaller than σ are removed. Yet, it is precisely this presmoothing which destroys image information such as the exact location of edges and corners. Ideally one would like a smoothing which does not destroy the edge information. This dilemma between smoothing of noise and the preservation of edges and corners has been commonly considered a "chicken and egg problem": An object of interest is more easily segmented, if one smoothes the grey value across the area corresponding to the object; however, in order not to smooth across the boundaries of the object, one already needs to know where the object is.

[5] Note that the Euclidean curve shortening flow by itself implicitly induces a shrinking of the contour. An appropriate choice of the constant c added to the curvature κ might help to compensate this effect.

[6] A precursor of such an implicit snake was proposed in [31, 121].

Interestingly, this chicken and egg dilemma can be tackled by a variational approach, which is described in the following section.

2.3.2 The Mumford-Shah Functional

In 1985, Mumford and Shah [135, 136] proposed to approximate a given input image f with a *piecewise smooth* function u by minimizing the functional

$$E(u, C) \;=\; \frac{1}{2}\int_{\Omega}(f-u)^2\,dx \;\;+\;\; \lambda^2\,\frac{1}{2}\int_{\Omega-C}|\nabla u|^2\,dx \;\;+\;\; \nu\,|C| \qquad (2.11)$$

simultaneously with respect to the image u and with respect to the contour C. The first term is a fidelity term, as it enforces that the function u is similar to the input image f in the L_2-sense. The second term enforces smoothness of the segmented image but permits discontinuities of u across a boundary denoted by C. The last term gives the one-dimensional Hausdorff measure of the length of this boundary. The parameter λ defines the spatial scale on which smoothing is done.

Similar models as (2.11) were formulated for the discrete case in a *Markov random field method* by Geman and Geman [81] and as the *weak membrane model* by Blake and Zisserman [18].

The free discontinuity problem in (2.11) triggered a large number of detailed theoretical studies (cf. [134, 117, 20]). Existence of global minimizers with a set C of closed boundaries was proved by Ambrosio [5] and de Giorgi et al. [61]. Regularity of the minimizing contours has been shown in [19, 6]. In [136], it is shown that corners or T-junctions are not permissible for minimizing contours, and that triple junctions can only arise with identical angles of 120°. For a detailed discussion of theoretical aspects we refer to the book of Morel and Solimini [134].

A coarse to fine method for minimizing the Mumford-Shah functional was proposed by Blake and Zisserman [18] under the name of *graduated non-convexity*. Essentially the authors convexify the original functional and determine a family of more and more non-convex approximations of the functional which are iteratively minimized, where the solution at each level serves as an initialization for the next (less convex) level. A similar coarse-to-fine approximation of the Mumford-Shah functional in terms of Γ-convergence was proposed by Ambrosio and Tortorelli [7]. Level set implementations of the Mumford-Shah functional were recently presented by Chan and Vese [35] and by Yezzi et al. [195].

A number of more heuristic methods of region growing (cf. [95]) can be considered precursors of the Mumford-Shah functional in the sense that they aim at partitioning the input image into piecewise homogeneous regions by appropriate hierarchical split and merge techniques.

2.3.3 Simplification and Probabilistic Generalization

If the parameter λ in equation (2.11) is increased, the smoothness constraint is given more weight. In the limit $\lambda \to \infty$, the approximation u will be forced to

be constant in each region $R_i \subset \Omega$ separated by the boundary set C:

$$u(x) = u_i \text{ for } x \in R_i, \tag{2.12}$$

The functional (2.11) then reduces to the *cartoon limit* [133, 136]:

$$E(u, C) \;=\; E(\{u_i\}, C) \;=\; \frac{1}{2} \sum_i \int_{R_i} (f - u_i)^2 \, dx \;+\; \nu \, |C|, \tag{2.13}$$

with the parameter ν appropriately rescaled.[7] Minimization of (2.13) results in an approximation of the input image f by a function u which is *piecewise constant* on a set of regions R_i separated by the boundary set C, where the constants u_i take on the mean grey value in each region R_i:

$$\frac{dE}{du_i} = 0 \quad \Longleftrightarrow \quad u_i = \frac{1}{|R_i|} \int_{R_i} f dx. \tag{2.14}$$

By associating with a given boundary set C the minimizing constants u_i in (2.14), the resulting functional reduces to a functional $E(C)$ which only depends on C. As discussed in [134, 136], the analysis of minimizers in terms of a finite number of rectifiable Jordan curves is drastically simplified in the piecewise constant case. We will not go into detail about these results, since we will later on further restrict permissible segmentations C to parametric closed contours. Results of minimizing the functional (2.13) by a pyramidal algorithm of recursive merging for the case of scalar (grey value) and vector-valued (texture) images were for example presented in [111].

Moreover, as detailed by Zhu and Yuille in their work on *region competition* [204], the piecewise constant Mumford-Shah functional provides an ideal starting point for a probabilistic interpretation of region-based segmentation. In contrast to the original Mumford-Shah functional (2.11), the simplified model (2.13) provides a segmentation for which the grey value in each region R_i is approximated by a constant u_i. Instead of approximating by a constant, one can more generally approximate the intensity in each region R_i by a probabilistic model $P(f(x)|\alpha_i)$ with a parameter vector α_i. The specific energy density for region R_i in the functional (2.13) is then replaced by the negative log-likelihood that a grey value f is encountered at point x, given the probabilistic model parameterized by α_i:

$$E(\{\alpha_i\}, C) = \; -\frac{1}{2} \sum_i \int_{R_i} \log P\big(f(x)|\alpha_i\big) \, dx \;+\; \nu \, |C|. \tag{2.15}$$

In this sense, the simplified Mumford-Shah model (2.13) corresponds to the specific case of Gaussian probability distributions for the grey values in the regions R_i with mean u_i and constant variance. As pointed out in [204], the obtained variational approach (2.15) is related to the approach of *minimum description length* [149, 118].

[7] As pointed out in [136], the functional (2.13) is equivalent to the *Ising model* [101], if it is discretized on a lattice and the constant values u_i for each region are restricted to $\{-1, +1\}$.

This probabilistic interpretation of the variational segmentation approach permits a number of extensions of the Mumford-Shah functional. In [204], the Gaussian probabilities are for example extended to permit for each region R_i not only a different mean u_i but also a different variance σ_i:

$$P(f(x)|\alpha_i) = \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\left(-\frac{(f-u_i)^2}{2\sigma_i^2}\right), \qquad \text{where } \alpha_i = \{u_i, \sigma_i\}. \tag{2.16}$$

The resulting segmentation process is then able to separate regions which have the same mean but different variances.[8]

As discussed in [204], the Gaussian model (2.16) is easily extended to vector-valued functions $f : \Omega \to \mathbb{R}^n$. This permits to segment images based on texture and color information.

An entirely different extension of the Mumford-Shah functional from the problem of grey value segmentation to that of *motion segmentation* is presented in Chapter 5. We named the resulting variational approach *motion competition*, because each region competes for the segmenting contour in terms of the log-likelihood that a given local motion estimate was generated from the respective motion model for this region.

Compared to the edge-based approaches discussed in Section 2.2, the Mumford-Shah functional essentially separates the two problems of modeling the image information in each region (by the function u) and the optimal positioning of the separating boundary (by the contour set C). The generalizations of the Mumford-Shah functional show some fundamental differences between region-based segmentation methods and the edge-based approach discussed earlier:

- The region-based segmentation does no longer rely on the vague concept of an *edge* — see the discussion in Section 2.1 — but rather maximizes a *homogeneity criterion* in each of the segmented regions.
- The region-based segmentation process can incorporate essentially arbitrary probabilistic models for the image information in the separate regions. This permits to elegantly treat very different segmentation cues such as image intensity, color, texture or motion in essentially the same probabilistic framework.
- Though both edge-based and region-based segmentation functionals tend to have several local minima for a given input image, we found in numerical implementations that for a large variety of segmentation tasks, the region-based formulation permits a convergence of the contour over much larger distances than commonly observed for edge-based approaches.

2.4 Diffusion Snakes

As discussed in the introduction, the goal of the present work is to introduce a prior knowledge on the expected shape of the contour into a segmentation

[8] However, during minimization the intensity variance at a particular point needs to be estimated over a window of a certain width which represents an additional parameter that must be optimized.

approach. Having briefly reviewed a number of variational approaches to segmentation, we will now present a modification of the Mumford-Shah functional which facilitates the introduction of a statistical prior on the shape of the segmenting contour.

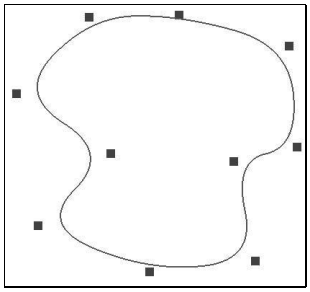


Figure 2.2: Example of a uniform quadratic B-spline curve. The control points are represented by squares.

2.4.1 Spline Representation

In order to model the distribution of a set of training shapes statistically, it is convenient to revert to explicit parametric descriptions of shape. We will focus on the shape variation of a single object. For simplicity we will therefore represent the segmenting contour C in (2.11) as a single closed spline curve of the form

$$C : [0,1] \longrightarrow \Omega\,, \qquad C(s) = \sum_{n=1}^{N} p_n\, B_n(s)\,, \tag{2.17}$$

where B_n are the uniform, periodic, quadratic B-spline basis functions [71] and $p_n = (x_n, y_n)^t$ denote the control points. This gives a compact representation of shape by a control point vector

$$z = (x_1, y_1, \ldots, x_N, y_N)^t, \tag{2.18}$$

with a continuous normal vector at each point of the contour — see Figure 2.2 for an illustration and Chapter 3 for more details.

The representation of the segmenting contour as a closed spline curve certainly restricts the class of possible boundary formations, not allowing open boundaries, contour splitting, etc. However, if the goal is to segment a simply-connected object of interest, then a restricted topology may be beneficial for the segmentation process. Moreover, the explicit contour permits the introduction of a statistical shape prior on the basis of the control point distribution associated with a set of training shapes. This will be discussed in more detail in Chapters 3 and 4.

2.4.2 Region-based Snakes

In a first spline-based implementation of the Mumford-Shah functional — see [57] — we represented the contour length in (2.11) as commonly done by:

$$|C| = \int_0^1 |C_s(s)|\, ds. \tag{2.19}$$

This produces a term proportional to the curvature in the evolution equation for the contour. In our framework of spline contours this term is not only computationally costly, but it also does not restrict the spline control points from clustering in one place. Once control points overlap, the normal vector on the contour becomes ill-defined. Since the contour is evolved along its normal, the segmentation process becomes instable. We found that this problem only arises in cases where the shape prior is absent, since otherwise the prior will restrict the control point polygon to a learnt distribution which was derived from training shapes with equidistant control points (cf. Chapter 3).

A modification of the original functional solves the problem of control point clustering: Replacing the original L_1-type norm (2.19) by a squared L_2-type norm, one obtains the the *diffusion snake* functional:

$$\textbf{(DS)} \qquad E(u, C) = \frac{1}{2}\int_\Omega (f-u)^2\, dx + \lambda^2\, \frac{1}{2}\int_{\Omega - C} |\nabla u|^2\, dx + \nu\, ||C||^2, \tag{2.20}$$

where

$$||C||^2 = \int_0^1 C_s^2\, ds \tag{2.21}$$

is the length constraint which is used for modeling curves known as *elastica*[9]. For a detailed discussion of the invariance properties associated with various smoothness functionals we refer to [63]. The internal energy (2.21) is also used for snakes, balloons and geodesic active contours — see equation (2.5). Therefore the diffusion snake model (2.20) can be considered a hybrid between the Mumford-Shah functional (2.11) and the snake (2.5). It is a region-based segmentation model with an explicit contour, having the external energy of the Mumford-Shah functional and the internal energy of a snake.

Minimizing the internal energy (2.21) with respect to C leads to an Euler-Lagrange equation of the simple form

$$C_{ss}(s) = 0 \quad \text{for } s \in [0, 1]. \tag{2.22}$$

For the quadratic B-spline curve this is equivalent to

$$p_i = \frac{p_{i-1} + p_{i+1}}{2}, \quad i = 1, \ldots, N. \tag{2.23}$$

[9] Other modifications of the Mumford-Shah functional with respect to length and curvature measures have been considered in [122].

Therefore, by minimizing (2.20), each control point p_i tends to be centered between its two neighbors. This is what makes (2.20) well suited for the spline-based implementation. Moreover, experimental results show that the common argument in favor of the L_1-type norm (2.19), namely that it allows discontinuities in the boundary, is of a more theoretical nature, since in our case a sufficiently fine parameterization allows the formation of arbitrarily sharp corners.

The same modification can be performed for the model (2.13) which gives the variational energy associated with the *simplified diffusion snake*:

$$\textbf{(SDS)} \qquad E(u,C) = \frac{1}{2}\sum_i \int_{R_i} (f-u_i)^2\,dx + \nu \int_0^1 C_s^2\,ds. \tag{2.24}$$

2.5 Minimization by Gradient Descent

The energies for the diffusion snake (2.20) and the simplified diffusion snake (2.24) are each simultaneously minimized with respect to both the segmenting contour C and the segmented image u.

2.5.1 Curve Evolution

Minimizing the diffusion snake functional (2.20) with respect to the contour C (for fixed u) leads to the Euler-Lagrange equation

$$\frac{\partial E}{\partial C} = \big[e^-(s) - e^+(s)\big] \cdot \boldsymbol{n}(s) - \nu\, C_{ss}(s) = 0 \quad \forall s \in [0,1]. \tag{2.25}$$

The terms e^+ and e^- denote the energy density [136]

$$e^{+/-} = (f-u)^2 \;+\; \lambda^2\,(\nabla u)^2 \tag{2.26}$$

outside and inside the contour $C(s)$, respectively, and $\boldsymbol{n}$ denotes the outer normal vector on the contour. For the simplified diffusion snake (2.24), u is piecewise constant and the second term in (2.26) disappears:

$$e^{+/-} = (f-u)^2. \tag{2.27}$$

Solving the minimization problem by gradient descent results in the evolution equation

$$\frac{\partial C(s,t)}{\partial t} = -\frac{\partial E(u,C)}{\partial C} = \big[e^+(s,t) - e^-(s,t)\big] \cdot \boldsymbol{n}(s,t) + \nu\, C_{ss}(s,t)\ \forall s, \tag{2.28}$$

where an artificial time parameter t has been introduced.

Equation (2.28) can be converted to an evolution equation for the control points by inserting the definition (2.17) of the contour as a spline curve:

$$\sum_{i=1}^N \frac{dp_i(t)}{dt} B_i(s) = \big[e^+(s,t) - e^-(s,t)\big] \cdot \boldsymbol{n}(s,t) + \nu \sum_{i=1}^N p_i(t) \frac{d^2 B_i(s)}{ds^2}. \tag{2.29}$$

This equation is now discretized with a set of nodes s_i along the contour to obtain a set of linear differential equations. The solution gives the temporal evolution for the coordinates of each control point (x_m, y_m):

$$\begin{aligned} \frac{dx_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \Big[\left(e_{s_i}^{+} - e_{s_i}^{-}\right) \boldsymbol{n}_x + \nu(x_{i-1} - 2x_i + x_{i+1}) \Big], \\ \frac{dy_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \Big[\left(e_{s_i}^{+} - e_{s_i}^{-}\right) \boldsymbol{n}_y + \nu(y_{i-1} - 2y_i + y_{i+1}) \Big]. \end{aligned} \tag{2.30}$$

The cyclic tridiagonal matrix $\mathbf{B}$ contains the spline basis functions evaluated at the nodes s_i: $B_{ij} = B_i(s_j)$, where s_i corresponds to the maximum of B_i.[10]

The two terms in the respective equations in (2.30) can be interpreted as follows: The first term maximizes the homogeneity in the adjoining regions as measured by the energy densities (2.26) or (2.27). This forces the contour towards the boundaries of the object. The second term minimizes the length (2.21) of the contour and thereby enforces an equidistant spacing of the control points.

2.5.2 Inhomogeneous Linear Diffusion

In order to minimize the modified Mumford-Shah functional (2.20) with respect to the segmented image u, we rewrite the functional in the following way:

$$E(u, C) = \frac{1}{2} \int_{\Omega} (f - u)^2 \, dx + \lambda^2 \frac{1}{2} \int_{\Omega} w_c(x) |\nabla u|^2 \, dx + \nu \, ||C||^2. \tag{2.31}$$

The contour dependence is now implicitly represented by an indicator function

$$w_c : \Omega \to \{0, 1\}, \qquad w_c(x) = \begin{cases} 0 & \text{if } x \in C \\ 1 & \text{otherwise} \end{cases}. \tag{2.32}$$

The Euler-Lagrange equation corresponding to this minimization problem is given by:

$$\frac{1}{\lambda^2} \frac{dE}{du} = \frac{1}{\lambda^2}(u - f) - \nabla \cdot (w_c \nabla u) = 0. \tag{2.33}$$

Its solution corresponds to the steady state of the following diffusion process:

$$\frac{\partial u}{\partial t} = \nabla \cdot (w_c \nabla u) + \frac{1}{\lambda^2} (f - u), \tag{2.34}$$

in which the contour enters as an inhomogeneous diffusivity defining a boundary to the diffusion process. This underlying diffusion process is what gave rise to the term *diffusion snake*.

[10] Rather than discretizing by a set of nodes, a similar set of linear differential equations is obtained by projecting equation (2.29) onto the basis functions $\{B_k\}_{k=1,\ldots,N}$. Although this solution is more elegant from a mathematical point of view, the obtained evolutions are not distinguishable from an experimental point of view and the computational overhead is somewhat larger, since the inverted matrix is not cyclic tridiagonal but cyclic pentadiagonal.

In the case of the cartoon limit, the diffusion process is replaced by an averaging process, such that the image u takes on the mean grey value u_i of each adjacent region R_i:

$$u_i \;=\; \frac{1}{|\Omega_i|} \int\limits_{\Omega_i} f \, dx. \tag{2.35}$$

These values are dynamically updated in alternation with the contour evolution.

Two different schemes have been used to approximate the diffusion process: A simple explicit approximation to the diffusion equation (2.34), and a more sophisticated multigrid scheme for solving the corresponding steady state equation (2.33). Both schemes are not straightforward because the strongly inhomogeneous coefficient function w_c has to be taken into account. Standard implementations may easily lead to diffusion across the discontinuity curve C and thus to undesired effects on the contour evolution. In the following, we will explain both schemes, starting with the simpler one.

A Simple Numerical Scheme

We approximate equation (2.34) by finite differences. Let $\tau > 0$ denote the step size in t-direction and let u_i^k be an approximation to $u(x,t)$ in some pixel i at $t = k\tau$. In a similar way, w_i^k and f_i serve as approximations to $w_c(x,t)$ and $f(x)$, respectively. Moreover, let $\mathcal{N}(i)$ denote the 4-neighborhood of pixel i. If we assume square pixels of size 1, a consistent discretization of (2.34) is given by

$$\frac{u_i^{k+1} - u_i^k}{\tau} \;=\; \sum_{j \in \mathcal{N}(i)} \sqrt{w_j^k w_i^k}\,(u_j^k - u_i^k) \;+\; \frac{1}{\lambda^2}\,(f_i - u_i^{k+1}) \qquad \forall\, i. \tag{2.36}$$

The proposed discretization of the indicator function w_c prevents diffusion across the curve C.

Assuming that u_i^k and its neighbors $\{u_j^k \,|\, j \in \mathcal{N}(i)\}$ are already known from the k-th iteration step, we can solve this equation explicitly for the unknown u_i^{k+1}:

$$u_i^{k+1} \;=\; \frac{\left(1 - \tau \sum\limits_{j \in \mathcal{N}(i)} \sqrt{w_j^k w_i^k}\right) u_i^k \;+\; \tau \sum\limits_{j \in \mathcal{N}(i)} \sqrt{w_j^k w_i^k}\, u_j^k \;+\; \frac{\tau}{\lambda^2} f_i}{1 + \frac{\tau}{\lambda^2}}. \tag{2.37}$$

This constitutes our simple iteration scheme for all pixels i and all iteration levels k.

Let us now investigate its stability. Equation (2.37) computes u_i^{k+1} as a weighted average of u_i^k, its four neighbors $\{u_j^k \,|\, j \in \mathcal{N}(i)\}$, and f_i. Note that the weights sum up to 1. Stability of this process can be guaranteed if all weights are nonnegative. Negative weights, however, can only appear in the first term, if τ is chosen too large. Since

$$0 \le \sqrt{w_j^k w_i^k} \le 1, \tag{2.38}$$

we end up with the stability restriction

$$\tau \leq \frac{1}{4}. \tag{2.39}$$

In this case we have a convex combination which guarantees that

$$\min_{j \in \mathcal{N}(i)} (f_j, u_j^k) \;\leq\; u_i^{k+1} \;\leq\; \max_{j \in \mathcal{N}(i)} (f_j, u_j^k) \qquad \forall\, i, k. \tag{2.40}$$

By initializing $u_j^0 := f_j$ and iterating over k, this simplifies to the discrete maximum-minimum principle

$$\min_{j \in \mathcal{N}(i)} f_j \;\leq\; u_i^k \;\leq\; \max_{j \in \mathcal{N}(i)} f_j \qquad \forall\, i, k. \tag{2.41}$$

This guarantees that the filtered image remains within the bounds of the original image.

A Multigrid Scheme for Diffusion Snakes

We discretize the steady state equation (2.33) by finite differences to obtain a linear system with natural (Neumann) boundary conditions:

$$\boldsymbol{A}u = f, \quad \text{and} \;\; \partial_n u = 0 \text{ on } \partial\Omega. \tag{2.42}$$

Solving this linear system with standard solvers like Gauss-Seidel or Jacobi takes a long time, as low frequencies in the error vanish slowly. Therefore we propose a multigrid implementation, which consists in recursively transferring the problem from a grid with size h to a coarser grid of size $2h$, and solving this to obtain a good initialization for the solution on the fine grid.

Note that a standard implementation of some numerical multigrid scheme, like the one in [171], may easily lead to a poor implementation of the steady state equation (2.33) due to the strongly inhomogeneous term w_c. The hierarchical representation of this term at multiple scales is even more difficult. For the diffusion snake to work, smoothing across the curve C must be prevented at all scales. The technical details of our implementation are described in Appendix B.

Results of the Multigrid Implementation

Figure 2.3 shows that using multigrid methods for solving the linear system (2.42) leads to a performance gain of several orders of magnitude compared to the use of standard algorithms. Using a w-cycle with three descending v-cycles (see Appendix B for details) and one step for presmoothing and postsmoothing on each level, one reaches the level of precision of a standard computer in only a few multigrid steps.

Analogous to the performance of standard solvers for common model problems, we found the computation time of the multigrid implementation to be fairly independent of the size of the smoothing parameter λ. This proves the robustness of our hierarchical scheme with respect to the strongly inhomogeneous diffusivity w_c. Moreover, the additional storage requirements are negligible. Further details can be found in Appendix B and in [177].

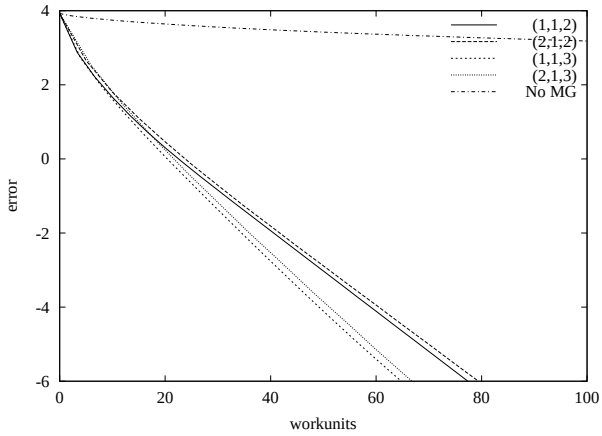


Figure 2.3: Comparison of different multigrid implementations and the symmetric Gauss-Seidel as a standard solver. The error is defined in logarithmic scale as $\log_{10} ||e||_2$. The respective numbers of presmoothing steps, postsmoothing steps and v-cycles on each level are given in brackets.

2.6 Numerical Results

In this section we present numerical results of image segmentations obtained with the diffusion snake (2.20) and the simplified diffusion snake (2.24) in the absence of a statistical shape prior.[11] The results demonstrate different properties of the diffusion snakes, showing their strengths and limitations.

The depicted contour evolutions correspond to various steps in the minimization of the diffusion snake functionals (2.20) or (2.24). Minimization is performed by iterating the evolution (2.30) of the contour C in alternation with an update of the smoothed approximation u, as defined in (2.34) or (2.35). The diffusion snake has two free parameters, namely the smoothing scale λ and the contour smoothness weight ν, whereas the simplified version only has the single parameter ν. For the description of the spline contour, we generally use a fixed number of 100 control points, apart from the two examples in Figures 2.4 and 2.5, where we used a larger number of 600 control points for a better resolution.

2.6.1 Separating Regions of Homogeneous Intensity

As discussed in the previous section, the contour of the diffusion snake evolves so as to maximize the grey value homogeneity in each region. Since minimization is done by gradient descent and since the functional is not convex, one expects the final segmentation to depend on the initialization.

This is demonstrated in Figure 2.4, where we segmented the same input image with the simplified diffusion snake for two slightly different initial contours. In the first evolution, the contour converges towards the black object, whereas for the second evolution, the contour converges towards the complement. This

[11] Internal energies such as the length constraint in (2.20) can be considered *shape priors* (cf. [195]), but they are purely geometric and do not introduce knowledge about a specific object of interest.

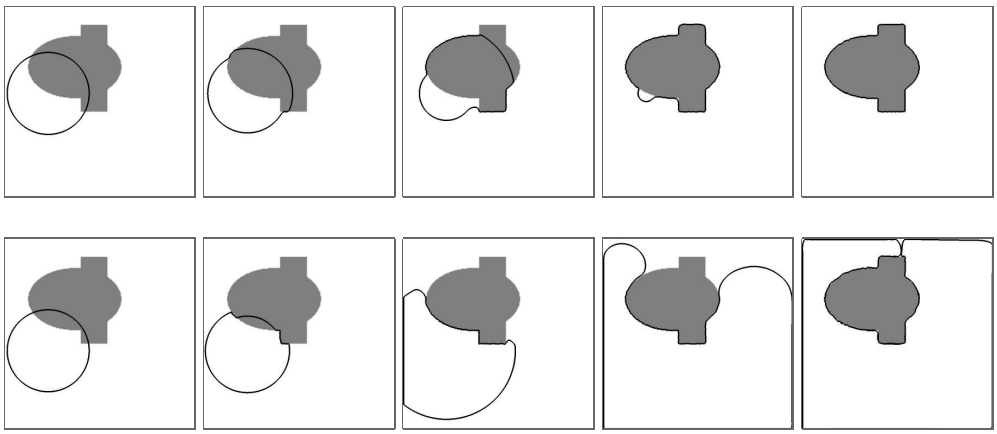


Figure 2.4: Contour evolution of the simplified diffusion snake for the same input image and two slightly different initializations. Due to the minimization by gradient descent, the final segmentation depends on the initial contour. The final contour on the bottom right shows that no contour splitting or merging mechanisms are incorporated in the segmentation process.

demonstrates the relatively simple mechanism underlying the contour evolution: If the mean grey value inside the initial contour is darker than the mean grey value outside the contour — see (2.35) — then the contour will evolve so as to encompass the dark area, and vice versa (apart from the contour smoothing induced by the length constraint).

2.6.2 Convergence over Large Distances

The example in Figure 2.4 showed a second fundamental property of the diffusion snake, namely that the contour can evolve over a fairly large spatial distance during the minimization process. Yet, at the same time the segmented structures are not blurred as for the classical snake — see Figure 2.1.

In many edge-based approaches, convergence over large distances is enhanced by a contraction or expansion force, as discussed for the balloon (2.7). Although such a term could be added to the diffusion snake functional, we did not do so, because it not only assumes a prior knowledge on whether the contour is to expand or to contract, but also tends to introduce a bias towards smaller or larger regions.

Since we did not include a balloon force in the models (2.20) and (2.24), the contour can both expand and contract without any change of parameters. This is demonstrated in Figure 2.5. For the same input image we performed a minimization on the functional (2.24), once with an initial contour which encompasses the object of interest (top row), and once with a contour which is mostly located inside the object (bottom row). The respective contour evolutions demonstrate that the diffusion snake can both contract and expand without any change in the parameter value.

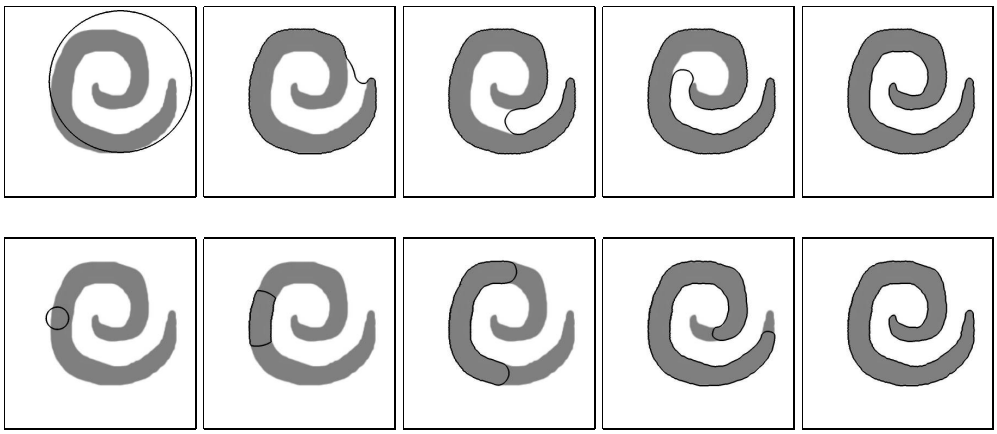


Figure 2.5: Inward and outward flow for the same parameter value. Since the diffusion snake models (2.20) and (2.24) do not contain a balloon term — see equation (2.7) — the contour can both expand and contract depending on the image information. During minimization the contour converges over a fairly large spatial distance without a particular bias on the size of the object of interest such as the balloon term.

2.6.3 Segmentation of Real-World Images

The Figures 2.4 and 2.5 show certain properties of the diffusion snake. Yet the example images are artificial ones. Figure 2.6, left side, shows a grey level image of a hand in front of a background and the initial contour (dashed line). The second image shows the final segmentation obtained with the diffusion snake model. Due to a large weight ν of the term minimizing the length of the contour, the thumb is cut off and the fingers are not fully segmented. If the parameter ν in (2.20) and (2.24) is decreased, the final contour is allowed to increase in length. The resulting segmentation is shown in the third image of Figure 2.6 for the cartoon limit. The hand is approximated better. However, some of the clutter in the background is included in the segmentation while the fingers are still not fully segmented.

The scene in Figure 2.6 contains little clutter, therefore segmentation results are rather good. Once the amount of clutter is increased, this changes considerably. Figure 2.7 shows an example of a hand in front of a strongly cluttered background. The grey value of the background is approximately the same as that of the hand. The result is that none of the segmentation approaches is able to extract the object of interest. Note that due to the underlying diffusion process the modified Mumford-Shah approach converges more locally than its cartoon limit, which simply segments areas of approximately constant grey value — see Figure 2.7, third image. This will be discussed in more detail in Section 3.6.4.

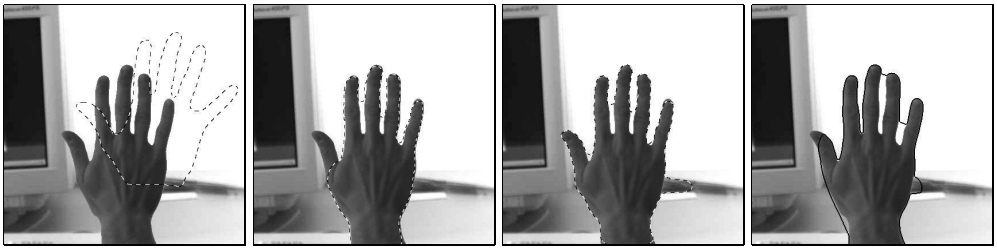


Figure 2.6: Segmentation with no prior.[12] From left to right: Initial contour, final segmentation for the diffusion snake, the simplified diffusion snake, and a level set implementation of geodesic active contours.

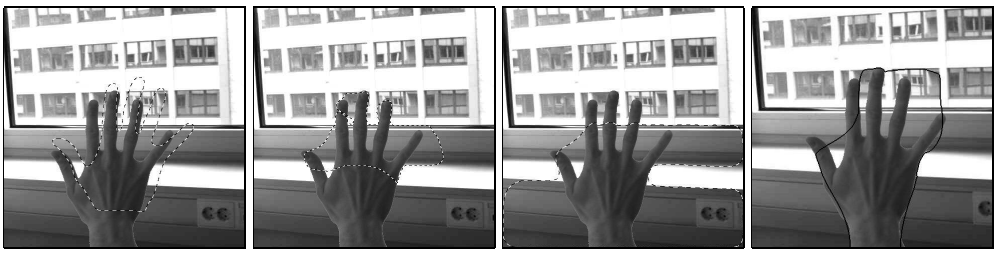


Figure 2.7: Segmentation with no prior[12] in strongly cluttered background. From left to right: Initial contour, segmentation results obtained for the diffusion snake, the simplified diffusion snake, and a level set scheme of geodesic active contours.

2.6.4 Comparison with Geodesic Active Contours

In order to compare our results to another segmentation approach, we performed a level set implementation of geodesic active contours — see Section 2.2.3. We opted for this comparison since the level set formulation of geodesic active contours is one of the most competitive among present segmentation methods. For the same input images f and the same initial contours C, we minimized the energy functional (2.8) for a Gaussian-smoothed input image f_σ, and the metric [185]

$$g(s^2) = \begin{cases} 1, & \text{if } s^2 = 0 \\ 1 - \exp\left(-\frac{3.315}{(s/\lambda)^8}\right), & \text{if } s^2 > 0 \end{cases} . \tag{2.43}$$

Here λ serves as contrast parameter.

We did not include any additional terms such as balloon forces since they assume a prior knowledge about whether the object of interest is inside or outside the initial contour. Moreover, the two diffusion snake models do not contain any such term either.

[12] For comparability with later results, the initialization corresponds to a hand shape, yet this does not have a relevant impact on the final segmentation.

Our geodesic active contour implementation used an efficient pyramid additive operator splitting (AOS) scheme that does not require to recalculate a distance transformation in each iteration [185].

The comparison in Figure 2.6 shows that the segmentation obtained by the Mumford-Shah based models and the one obtained by the geodesic active contour model are similar for homogeneous background. However, the comparison with Figure 2.7 indicates that in a strongly cluttered background the geodesic active contours give a more satisfactory approximation of the object of interest — indicating at least its approximate location.

One should however keep in mind, that the model formulations are conceptually very different: Whereas the geodesic active contour model is directly governed by the gradient of the smoothed input image, this is different for the Mumford-Shah model — especially for the case of the cartoon limit, which is a region-based rather than an edge–based segmentation approach.

Moreover, in the case of the geodesic active contour model, the final contour is obtained as the zero level set of a higher dimensional surface. In our model formulation the final segmentation curve is obtained in form of a parameterized spline curve. The latter permits a straight-forward implementation of shape statistics and similarity invariance — see Chapters 3 and 4. Moreover, an explicit representation of the segmented contour is of interest in terms of generative-model-based vision.

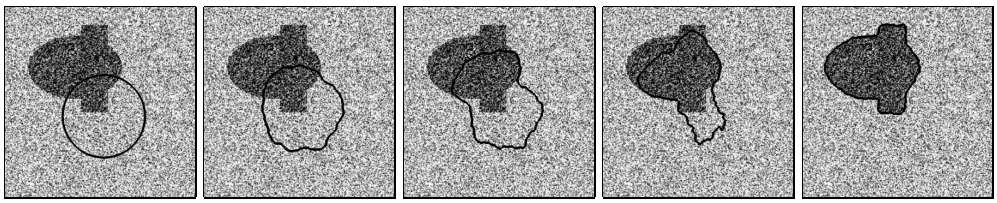
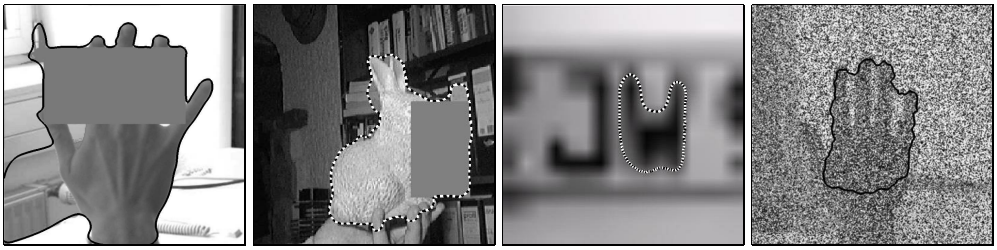


Figure 2.8: Segmentation of the input image shown in Figure 2.4 which was corrupted by noise. 60% of the image pixels were replaced by an arbitrary grey value sampled from a uniform distribution over the interval $[0, 255]$.

2.6.5 Robustness to Noise

In many practical computer vision applications, the assumptions about the grey value homogeneity of object and background are not fulfilled. The image may for example be heavily corrupted with noise. Edge-based approaches tend to tackle this difficulty by presmoothing the image. As discussed in Section 2.2, this tends to destroy a lot of valuable image information such as the exact location of edges. Moreover, this approach assumes some prior knowledge about the spatial scale of the noise which determines the optimal smoothing scale.

This is different for region-based approaches such as the diffusion snakes, where the problems of smoothing and optimal edge location are separated by two variables u and C in the functionals (2.20) and (2.24). Figure 2.8 demonstrates that the resulting segmentation process is robust to noise without blurring edge information.



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

Figure 2.9: Corrupted low-level intensity cues. Segmentations obtained by the simplified diffusion snake for an occluded hand **(a)**, for the occluded figure of a rabbit **(b)**, for a subsampled and smoothed image of a license plate section **(c)**, and for a noisy image of a hand **(d)**. The noise was induced by replacing 75% of the pixels with an arbitrary grey value sampled from a uniform distribution over the interval $[0, 255]$.

2.6.6 Limitations of Purely Image-based Segmentation

As a motivation for the following chapters, we will now present a number of results which emphasize the need to incorporate into the segmentation process some prior information on the objects of interest.

The diffusion snake model produces segmentations based on a simple criterion for the low-level image intensity information. As for any purely image-based segmentation approach, the contour will inevitably fail to converge towards the desired segmentation as soon as the assumptions about the low-level intensity statistics are no longer fulfilled. In our case a number of reasons may induce such a failure:

- If there are large amounts of *clutter*, then the hypothesis of homogeneous background intensity may be strongly violated such that the final segmentation fails to capture the object of interest. This is shown by the results in Figure 2.7.

- The object of interest may be partially *occluded*, such that the desired segmentation is neither defined in terms of homogeneous grey value nor in terms of well-defined edge information. Two example images and the resulting segmentation for the simplified diffusion snake are shown in Figure 2.9, (a) and (c).

- The image information may be insufficient due to *subsampling* or *coarse graining*. This problem commonly arises in practical applications with cameras of low resolution. Figure 2.9, (c) shows a segmentation result for a subsampled and smoothed section of a license plate.

- If the input image is strongly corrupted by *noise*, then the low-level intensity information may be so perturbed that it will not drive the contour towards the desired segmentation. Such an example is shown in Figure 2.9, (d).

In all of these cases, the low-level image information is not sufficient to define the desired segmentation. Yet, for the human observer, the objects are clearly visible. As argued in Chapter 1, the human visual system tends to rely on both the low-level image information and higher-level concepts about objects which are familiar from a previous learning process.

How such an interaction of low-level intensity cues and high-level knowledge about the shape of expected objects can be combined in a segmentation process on the basis of the variational approaches (2.20) and (2.24) will be presented in the next two chapters. In particular, we will show that including statistical shape knowledge in the variational approach permits to obtain the desired segmentation for the examples in Figure 2.9.

Chapter 3

Linear Shape Statistics in Segmentation

In this chapter, we present an extension of the diffusion snake models **DS** (2.20) and **SDS** (2.24) by a term which favors the formation of contours which are familiar from a previous learning process.

In Section 3.1, we detail the representation of the contour as a spline curve. We discuss methods of measuring the distance between two contours. We present a method of automatic contour extraction and introduce a method of aligning a set of training contours both with respect to similarity transformations[1] and with respect to cyclic permutation of the control points.

In Section 3.2, we briefly review the method of principal component analysis (PCA), which extracts the modes of largest variation. In Section 3.3, we present a model of linear shape statistics which is based on the assumption that the control point vectors corresponding to the set of training shapes are distributed according to a Gaussian distribution. Compared to PCA, we do not work in the subspace spanned by the first few eigenmodes, but rather in the full space of possible spline contours. Advantages and disadvantages of such a probabilistic embedding are discussed. Moreover, we compare two approaches for regularizing the sample covariance matrix.

In Section 3.4, we discuss the problem of integrating invariance under certain transformation groups into the shape energy. In particular, we present two complementary approaches: Firstly, we investigate the possibility of *learning* invariances from the set of sample contours. We show that this works for translation, but that it cannot be extended to similarity transformations. Secondly, we present a closed-form solution for a variational integration of similarity invariance, which is based on the spline representation of the contour. Compared to most other approaches, it does not require the introduction of explicit parameters to account for translation, scaling and rotation. In this context, we discuss two alternative approaches to introduce invariance, namely the optimization of explicit pose parameters and the use of intrinsically invariant shape descriptors.

[1] In this work, the group of similarity transformations only encompasses the *direct* similarities [126] rotation, scaling and translation — mirroring will not be considered.

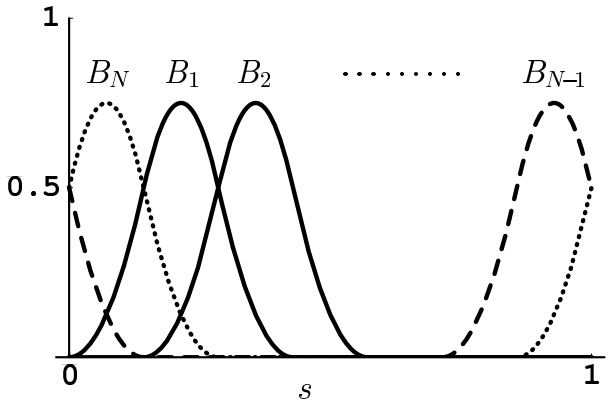


Figure 3.1: Periodic, quadratic and uniform B-spline basis functions on the interval $[0, 1]$.

In Section 3.5, we detail how the Gaussian shape prior is incorporated in the variational approach. Section 3.6 contains experimental results of the diffusion snake model with the linear shape prior. These are chosen so as to highlight different aspects of the obtained segmentation method.

3.1 Shape Learning

3.1.1 Shape Representation

We represent the silhouette of an object by a closed curve of the form

$$C_z : [0,1] \longrightarrow \Omega \subset \mathbb{R}^2\,, \qquad C_z(s) = \sum_{n=1}^{N} p_n\, B_n(s)\,, \tag{3.1}$$

where B_n are the uniform, periodic, quadratic B-spline basis functions [71, 17] shown in Figure 3.1, and $p_n = (x_n,\, y_n)^t$ denote the control points. To simplify the notation, we denote the vector of all control points by

$$z = (x_1, y_1, \ldots, x_N, y_N)^t. \tag{3.2}$$

In practice the number N of control points is fixed to a value which permits sufficient contour detail. See Figure 3.2 for an illustration.

We decided for a quadratic spline representation for several reasons: First, compared to a polygonal representation the contour in (3.1) is differentiable at all points $s \in [0, 1]$ such that a normal vector is easily defined. Such a normal vector is crucial since contour evolution approaches generally propagate the contour along its normal vector. Secondly, the smooth contour permits a compact representation of many natural shapes. And thirdly, the computational overhead due to the quadratic basis functions is moderate. Yet other representations might be interesting to study. In particular, some speed-up can be expected of a polygonal representation. Since in practice the above representation gives a sufficiently high shape variability, we did not see a need for more

Input image 20 points 40 points 70 points 100 points

Figure 3.2: Spline representation of a hand shape (left) with increasing resolution.

elaborate representations such as nonuniform or rational splines. Such representations introduce a higher number of parameters and additional mathematical complexity.

For an efficient drawing of the spline curve (3.1), we implemented the algorithms of de Boor and de Casteljau, where the latter one works on the equivalent Bézier representation of (3.1). We refer to [96] for details.

3.1.2 Shape Metrics

An important issue of shape statistics is that of defining appropriate metrics. Given two curves C_z and $C_{\hat{z}}$ of the form (3.1), what is the distance between them? A sensible analytical solution is given by:

$$||C_z - C_{\hat{z}}||^2 := \min_g \int_0^1 \big(C_z(s) - C_{\hat{z}}(g(s))\big)^2 \, ds, \tag{3.3}$$

where minimization is done over all continuous monotonous reparameterizations $g : [0,1] \to [0,1]$. Since the optimization over all possible reparameterizations is quite tedious, we will in practice revert to approximations of the above minimization.

Assuming that the correct reparameterization is fairly close to the identity, one can perform a local optimization [17] by enforcing that the derivative of the integrand with respect to g should be zero for all $s \in [0,1]$. A Taylor expansion leads to the approximation:

$$||C_z - C_{\hat{z}}||^2 \approx \int_0^1 \big[\big(C_z(s) - C_{\hat{z}}(s)\big) \cdot \boldsymbol{n}(s)\big]^2 \, ds,$$

where $\boldsymbol{n}$ is the normal vector on one of the curves. Thus the integrand measures only the displacement in direction of the normal. If both curves are sufficiently close to a reference curve, then the normal can be taken on the reference curve.

For our purpose we reverted to a simpler (and rougher) approximation of the distance measure (3.3). Given two spline curves C_z and $C_{\hat{z}}$, both parameterized

with N control points, we approximate:

$$||C_z - C_{\hat{z}}||^2 \approx \min_{\pi} \int_0^1 \big(C_z(s) - C_{\pi\hat{z}}(s)\big)^2 \, ds,$$

where minimization is done over the set of cyclic permutations (renumberings) π of the N control points in the vector $\hat{z}$. Since the control points are set (more or less) equidistantly, this approximation can be improved by working with larger numbers of control points.

Once the correct parameterization is determined and enforced, i.e. the control points are appropriately renumbered, the distance can be expressed in terms of the control point vectors:

$$d(C_z, C_{\hat{z}}) = \int_0^1 \big(C_z(s) - C_{\hat{z}}(s)\big)^2 \, ds = (z - \hat{z})^t \, A \, (z - \hat{z}), \tag{3.4}$$

with the matrix A given by

$$A = B \otimes \mathrm{I}_2, \tag{3.5}$$

where I_2 is the 2×2 unit matrix, $\otimes$ denotes the Kronecker product and the matrix B contains the overlap integrals of the spline basis functions:

$$B_{ij} = \int_0^1 B_i(s) B_j(s) \, ds \quad \forall \, i, j = 1, \ldots, N. \tag{3.6}$$

As the matrix A is symmetric and positive definite, it induces a scalar product on the space of spline curves:

$$(C_z, C_{\hat{z}}) := z^t \, A \, \hat{z} \tag{3.7}$$

In this way we obtain a Hilbert space structure for the space of spline curves. In fact we can embed the contour C_z in a Euclidean vector space by associating each curve C_z with a vector $A^{1/2} \, z$, with the control point vector z given in (3.2) and the matrix A from (3.5).

For the quadratic spline basis functions shown in Figure 3.1, the matrix B in (3.6) is cyclic pentadiagonal with a dominant diagonal. For computational simplicity we will neglect the overlap by approximating $B_{ij} \approx \delta_{ij}$ which implies that $A \approx \mathrm{I}$. This means that we approximate the Mahalanobis distance in (3.4) by the simpler Euclidean distance between the control point polygons

$$d(C_z, C_{\hat{z}}) \approx (z - \hat{z})^t \, (z - \hat{z}). \tag{3.8}$$

By this approximation we associate each contour with the corresponding control point polygon, and the distance between two contours is measured in terms of the distance between the two control point polygons. Although this appears to be a fairly rough approximation, we will see that it drastically simplifies future modeling. Moreover, with an increasing number of control points, the control point polygon will approach the contour as shown in Figure 3.2. For a further justification of this approximation we refer to Appendix A, where we will discuss the effect of this approximation on shape alignment and shape statistics in more detail.

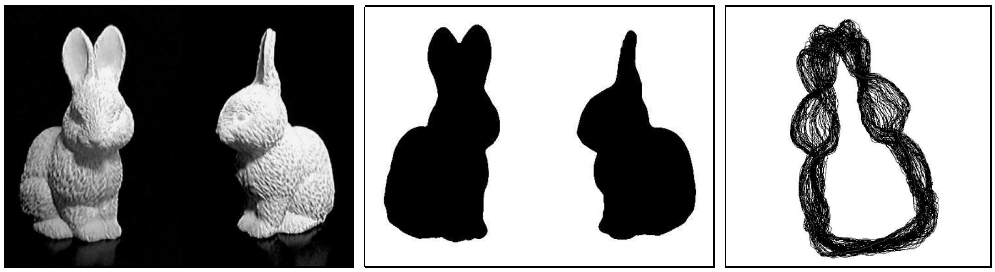


Figure 3.3: Example views, binarized training images and extracted contours after alignment.

3.1.3 Automatic Shape Acquisition

In order to learn the silhouettes for a set of objects or different views and poses of an object, one possible way is to manually place certain landmark points along the object outline on all the views in the training set. This has been done for example in [44]. However, this is a supervised approach and the manual interaction can be cumbersome if the number of training shapes is large. An application to large training sets is therefore infeasible.

Therefore, we have chosen a different approach [10], where the training images are preprocessed in order to adapt them to an automatic spline fit. In practice, we binarize the training images by applying a threshold and — if necessary — a median filter to remove noise. In more difficult background conditions one can preprocess the input images by background subtraction. Then we extract the boundary of the object in form of a chain code of length m: $\{f_n \in \Omega\}_{n=1,\ldots,m}$. The optimal spline control point polygon is now given by:

$$z^{x/y} = B^{-1} \frac{1}{m} \sum_{n=1}^{m} \boldsymbol{B}\left(\tfrac{n}{m}\right) f_n^{x/y},$$

where the superscripts x/y refer to the vectors of x- and y-components, respectively. $\boldsymbol{B}$ is the vector of basis functions $\{B_n\}$, and B is the matrix (3.6) of overlap integrals. Figure 3.3 shows an example of 100 training contours extracted from binarized example views of a 3D object.

Compared to the manual selection of landmark points, the automatic extraction of spline control points does not provide us with the correct correspondences of control points for several shapes. For example, in the process of manual landmark selection for a set of hand shapes, one can guarantee that all fingertips of one hand shape will be associated with the corresponding ones of the other hand shapes. For the automatic acquisition process described above, this can no longer be guaranteed. In practice we solve this problem by determining the cyclic renumbering of control points which produces the best alignment of the two contours, as explained in the next section. Certainly this solution is suboptimal in several ways: First, there may be no control point at the corresponding contour position, e.g. the corresponding fingertip. This limitation becomes negligible if a sufficient number of control points is used. Secondly, if

the two shapes strongly different in one location — for example if one of the fingers in one hand image is much longer than in the other images — then due to the equidistant spacing of control points the correct correspondences of all subparts cannot be enforced. We will come back to this limitation at the end of this work in Section 6.2.

3.1.4 Alignment of Training Contours

Given a set of training contours, we want to eliminate the degrees of freedom corresponding to cyclic permutation of control points and those corresponding to similarity transformations, i.e. translation, rotation and scaling.

For the purpose of alignment of contours, we will deviate from the standard notation (3.2) and identify each control point vector $z \in \mathbb{R}^{2N}$ with a vector $z \in \mathbb{C}^N$:

$$z = (x_1 + iy_1, \ldots, x_N + iy_N)^t, \quad \text{where } i = \sqrt{-1}.$$

Let z, $\hat{z} \in \mathbb{C}^N$ be two such control point vectors. For the beginning we will assume that control points are numbered such that we already have the correct correspondence. Let both vectors be centered:

$$z^t\, 1_N = \hat{z}^t\, 1_N = 0, \quad \text{where } 1_N = (1, \ldots, 1)^t \in \mathbb{R}^N.$$

The optimal alignment or superimposition of z and $\hat{z}$ with respect to the similarity transformations is called the *full Procrustes fit* [67, 83]. Performing this fit amounts to minimizing the distance

$$D^2(z, \hat{z}) = ||z - \alpha\hat{z} + \beta||^2, \tag{3.9}$$

with respect to the translation $\beta \in \mathbb{C}$ and the parameter $\alpha = re^{i\phi} \in \mathbb{C}$ which accounts for scaling by a constant $r \geq 0$ and rotation by an angle $\phi \in [0, 2\pi]$. Setting the corresponding derivatives to zero, one can solve for the minimizing parameters [67, 190] to obtain:

$$\beta = 0, \quad \alpha = \frac{\hat{z}^* z}{\hat{z}^* \hat{z}}, \tag{3.10}$$

where $*$ denotes transposition and complex conjugation. If the initial vectors z and $\hat{z}$ are centered and also normalized (i.e. $z^* z = 1$), then the distance (3.9) with the optimal parameters (3.10) is called the *full Procrustes distance*. It is given by:

$$\hat{D}^2(z, \hat{z}) = 1 - |z^* \hat{z}|^2. \tag{3.11}$$

Given a set of training vectors $\chi = \{z_i \in \mathbb{C}^N\}_{i=1,\ldots,m}$ which are centered and normalized, we can align them in several ways. One way is to align them all by the above approach, minimizing the full Procrustes distance with respect to one of the vectors (say the first one). Rather than distinguishing one particular vector as the reference vector, one can instead align them with respect to the Procrustes estimate of the mean vector which is defined as

$$\hat{\mu} = \arg\inf_{\mu} \sum_{i=1}^{m} \hat{D}^2(z_i, \mu).$$

As shown in [104], there is a closed-form solution for this Procrustes mean. In fact, using the definition (3.9) of the Procrustes distance and the solution (3.10) for the optimal transformation parameters, one obtains:

$$\hat{\mu} = \arg\inf_{\mu} \sum_{i=1}^{m} \left(1 - \frac{\mu^* z_i z_i^* \mu}{\mu^* \mu}\right) = \arg \sup_{||\mu||=1} \mu^* S \mu.$$

The solution of this expression is given (up to rotations) by the complex eigenvector $\hat{\mu}$ corresponding to the largest eigenvalue of the matrix

$$S := \sum_{i=1}^{m} z_i \, z_i^*.$$

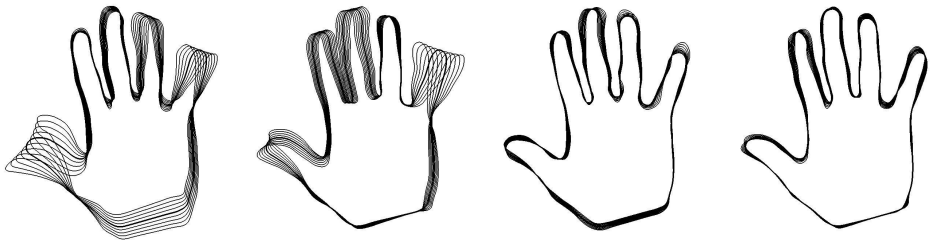
The vectors $\{z_i\}$ are then aligned with respect to this full Procrustes mean $\hat{\mu}$ as shown above. This simultaneous superimposition of *several* shape vectors is called *generalized Procrustes analysis* [84].

Since we are working with automatically extracted shape vectors z_i, the correct correspondences between control points are not given. Therefore we have to include an optimization with respect to *cyclic renumbering of control points* in the above alignment process. There are N possible cyclic renumberings for each shape in the set. Carrying out the above method of generalized Procrustes analysis for all possible N^{m-1} combinations of renumberings of the training vectors would be far too time-consuming.

An iterative solution to the generalized Procrustes analysis, when the correct correspondences are given, is used in [41]. This method can be extended by an additional optimization over cyclic renumbering of the control points. The resulting alignment procedure is as follows:

- The training vectors are centered and normalized. One of them is chosen as an initial estimate of the mean.
- The others are aligned to the estimate of the mean with respect to similarity transformations and renumbering. The correct renumbering is obtained by minimizing the full Procrustes distance (3.11) over all N cyclic permutations. Then the correct alignment parameters are given by equation (3.10).
- Once all shape vectors are aligned to the estimate of the mean and normalized afterwards, the mean is updated. Then the entire process is iterated.

Although we did not find any proof of convergence, we found the iteration to converge after a few steps in practice. The normalization of the shape vectors in each iteration is of importance because otherwise, the shape vectors tend to shrink with every iteration. This is due to the fact that the Procrustes fit of a shape vector z to the reference vector $\hat{z}$ shrinks the vector z by the factor $|\alpha| \leq 1$ — see equation (3.10) — where $|\alpha| = 1$ if and only if the two vectors are the same.



1st p.c. 2nd p.c. 3rd p.c. 4th p.c.

Figure 3.4: Sampling up to two standard deviations along the first four principal components from the mean for a set of 15 hand shapes.

3.2 Principal Component Analysis

There is a number of ways to statistically analyze the shape variation contained in a set of training shapes. The present section focuses on linear statistics, where the term *linear* indicates that all permissible shapes are given by the mean shape plus a linear combination of a set of eigenmodes. The eigenmodes which capture most of the shape variation encountered in the training set are called the *principal components* or *eigenshapes*. Principal component analysis (PCA) is a standard technique. Its first application to modeling shape variation was proposed in [45] for polygonal outlines under the name of *point distribution model*. A subsequent extension to spline curves was presented in [10]. We will briefly sketch this method now.

Let $\chi = \{z_i \in \mathbb{R}^{2N}\}_{i=1,\ldots,m}$ be a set of training shapes, aligned as discussed in Section 3.1.4. Denote the sample mean by:

$$\bar{z} = \frac{1}{m} \sum_{i=1}^{m} z_i, \tag{3.12}$$

and the (unbiased) sample covariance matrix by:

$$\Sigma = \frac{1}{m-1} \sum_{i=1}^{m} (z_i - \bar{z})(z_i - \bar{z})^t. \tag{3.13}$$

The symmetric real matrix Σ can be diagonalized. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ be the ordered (nonnegative) eigenvalues of Σ. The modes of largest variation are given by the eigenvectors e_i corresponding to the largest eigenvalues λ_i. The eigenvalue λ_i is a measure of the relative amount of the total variation explained by the eigenmode e_i. A compact lower-dimensional shape model is given by linear combinations of such eigenmodes added to the mean shape:

$$z(\alpha_1, \ldots, \alpha_r) = \bar{z} + \sum_{i=1}^{r} \alpha_i \sqrt{\lambda_i}\, e_i, \qquad \text{where } r < m. \tag{3.14}$$

Scaling by $\sqrt{\lambda_i}$ has been introduced for normalization; it corresponds to the standard deviation in the eigendirection e_i. The shape variation associated

with a given eigenmode can be visualized. Figure 3.4 shows a sampling along the first four eigenmodes from the mean.

Extending the *shape parameters* $\{\alpha_i\}$ by *pose parameters* $\{x, y, s, \theta\}$ to account for translation, scaling and rotation of a given shape, one obtains the *Active Shape Model* [47]. In general, this small number of parameters is locally or globally optimized to fit features in an image.

3.3 The Gaussian Model in Shape Space

3.3.1 From Learnt Shape Statistics to a Shape Energy

The model of shape statistics we propose in this section is based on the PCA approach, the differences are detailed in the following. We assume that the training shapes $z_i \in \chi$ — aligned as discussed in Section 3.1.4 — are distributed according to a multivariate Gaussian distribution. The main axes of the associated hyperellipsoid are given by the principal components. Yet we do not restrict the contours to the linear subspace spanned by the first few of these eigenmodes. Although we sacrifice the computational efficiency induced by such a sparse representation, there are several reasons why we decided to work in the full space of possible spline curves:

- The "full" Gaussian model is more faithful from a probabilistic point of view: Given a finite number of training shapes, the modes orthogonal to the first few principal components should not be assigned a zero probability.

- By not limiting the degrees of freedom of the evolving contour to the subspace spanned by a small number of eigenmodes, we can perform a direct comparison of segmentation processes *with* and *without* the statistical prior, without further modifications of the model.

- By dropping the notion of *eigenmodes*, a generalization to improved statistical models such as the one proposed in Chapter 4 is more straightforward. In fact, for more elaborate nonlinear models of shape variation, a description in terms of eigenmodes is no longer possible.

We assume that the training shapes $z_i \in \chi$ are distributed according to a Gaussian probability density:

$$\mathcal{P}(z) \propto \exp\left(-\frac{1}{2}\,(z-\bar{z})^t\,\Sigma_\perp^{-1}\,(z-\bar{z})\right),$$

with the sample mean $\bar{z}$ as defined in (3.12).

If the training shapes span a lower-dimensional subspace of the $2N$-dimensional input space, then the sample covariance matrix Σ from (3.13) is not invertible. Therefore we perform a regularization of the form:

$$\Sigma_\perp = \Sigma \;+\; \lambda_\perp\,\left(I - V\,V^t\right), \tag{3.15}$$

where V is the matrix of eigenvectors of Σ. In this way, we replace all zero eigenvalues of the sample covariance matrix Σ by a constant

$$\lambda_\perp \in [0, \lambda_r], \tag{3.16}$$

where λ_r denotes the smallest non-zero eigenvalue of Σ.[2]

Maximizing the shape probability $\mathcal{P}(z)$ is equivalent to minimizing its negative logarithm. Up to a constant the latter is given by the quadratic *shape energy*

$$E(z) = \frac{1}{2}\,(z-\bar{z})^t\,\Sigma_\perp^{-1}\,(z-\bar{z}). \tag{3.17}$$

This *Mahalanobis distance* measures the dissimilarity of a given shape z with respect to a set of training shapes which are encoded in terms of their second-order statistics by the sample mean $\bar{z}$ and the (appropriately regularized) sample covariance matrix.

3.3.2 On the Regularization of the Covariance Matrix

Regularizations of the covariance matrix as done in (3.15) were proposed by several authors under the names of *residual variance approximation* and *sensible* or *probabilistic PCA* [42, 131, 155, 176, 57]. Commonly [131, 176] the constant $\lambda_\perp$ is estimated as the mean of the replaced eigenvalues by minimizing the Kullback-Leibler distance of the corresponding densities:

$$\lambda_\perp = \frac{1}{2N-r}\sum_{i=r+1}^{2N}\lambda_i. \tag{3.18}$$

In our case, N is the number of spline control points. However, we believe that this is not the appropriate regularization of the covariance matrix in our situation. The Kullback-Leibler distance is supposed to measure the error with respect to the correct density, which means that the covariance matrix calculated from the training data is assumed to be the correct one. But this is not the case because the number of training points is limited. In particular, setting $\lambda_\perp$ to the mean of the replaced eigenvalues would not solve the problem in our case, since the replaced eigenvalues are all zero.

A conceptually sound approach to determine the optimal mean $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$ for a given set of sample points $\{z_i\}$ is the Bayesian maximum a posteriori estimate:

$$\{\tilde{\mu}, \tilde{\Sigma}\} = \arg\max_{\mu,\Sigma} P\left(\{z_i\}|\mu,\Sigma\right)\, P\left(\mu,\Sigma\right).$$

However, this approach assumes knowledge about the prior probability distribution $P\left(\mu,\Sigma\right)$ on the space of all Gaussian distributions, which means that one needs to specify a model for the likelihood of different means and covariance matrices. Making assumptions about this prior distribution is probably no more

[2] We point out, that the inverse $\Sigma_\perp^{-1}$ of the regularized covariance matrix defined in (3.15) fundamentally differs from the so-called pseudoinverse.

effective than making a direct assumption about the value of the regularizing constant $\lambda_\perp$.

An approach which is very similar to (3.15) is that of *regularized discriminant analysis* [76]. There a regularized covariance matrix is obtained by a linear combination of the sample covariance matrices associated with an *isotropic* and an *anisotropic* Gaussian model:

$$\Sigma_\gamma = \gamma\Sigma + (1-\gamma)\sigma^2 I,$$

with a regularization parameter $\gamma \in [0,1]$, and σ^2 being the variance of the sample data (cf. [89]).

With the constraint (3.16) we obtain a probabilistic model for which unfamiliar variations from the mean are less probable than the smallest variation observed on the training set. This appears to be a reasonable hypothesis. Lacking a more precise estimate, we fix

$$\lambda_\perp = \lambda_r/2$$

in all applications.

3.3.3 On the *Curse of Dimensionality*

This "conservative" hypothesis about the probability of unfamiliar shape deformations permits us to work with sample sizes which are much smaller than the dimension of the underlying vector space. The commonly employed expression *curse of dimensionality* [12] suggests that the number of samples needed to obtain reliable statistics increases rapidly with the dimension of the input data. However, for several reasons the suggested limitations do not apply in our case:

- The full dimension of the vector space is not necessarily relevant. For example, if all training data is confined to a low-dimensional subspace, the relevant dimension is obviously that of the subspace.

- The regularization (3.15) permits to associate a sensible probability even with directions which are orthogonal to the subspace spanned by the training data. Obviously, this regularization becomes more important for smaller sample sizes.

- In practice the training shapes are usually not sampled completely at random. For the acquisition of training shapes, we generally sample a set of more or less *representative* views of a given object, such that we can expect more reliable estimates of the mean and the covariance matrix for a given number of sample shapes.

Therefore, the eigenmodes as shown in Figure 3.4 do not change much whether we use 6, 15 or 50 training shapes — although the input dimension is 200 (for 100 control points).

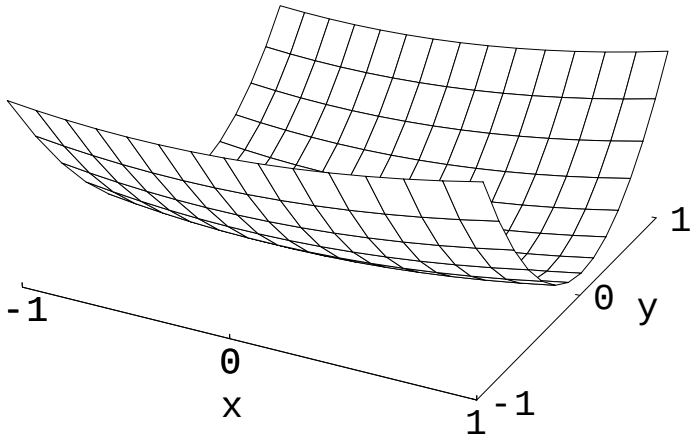


Figure 3.5: Schematic plot of the quadratic shape energy (3.17). Familiar deformation modes are represented by x, unfamiliar ones by y (see text).

3.3.4 The Elastic Tunnel of Familiar Shapes

To visualize the effect of the regularizing constant $\lambda_\perp$ in the definition of $\Sigma_\perp$, Figure 3.5 shows a schematic plot of the shape energy. For the purpose of clarity the shape space is reduced to two dimensions — a learnt (familiar) direction x (representing the subspace of the principal components) and an orthogonal direction y. The shape energy is less sensitive to shape deformation along the learnt directions than to deformation in orthogonal directions. The entire space of permissible shape variations is basically reduced to an *elastic tunnel of familiar shapes*. Restricting the shape variability to this tunnel amounts to a drastic reduction in the effective dimensionality of the search space. Yet, in contrast to the PCA approach, deformation in "unfamiliar" directions is still possible.

3.4 Incorporating Invariance

By construction, the shape energy (3.17) is not invariant with respect to transformations such as translation, rotation and scaling of the input vector. Yet, certain invariances of the shape prior are desirable in many applications, for example in a recognition task one might want the computer to accept a certain shape as a hand, independent of its rotation and translation.

In Sections 3.4.1 and 3.4.2, we will discuss two complementary approaches to tackle this problem. The first one is an approach of learning invariance, which induces a *robustness* to certain transformations derived from the training shapes. Although it works well for translation only, we will nevertheless present it here because it is a straight-forward extension of the learning process presented in the previous sections. The second approach is complementary in the sense that the invariance is not *learnt implicitly* from the examples but rather *constructed explicitly*. In particular, we present a shape energy which is by definition invariant with respect to similarity transformations. We derive a

gradient descent equation on this energy, and we discuss extensions to affine invariance.

In Section 3.4.3, we discuss some commonly used alternatives to incorporate invariance, namely the minimization of *explicit* pose parameters and the use of inherently invariant shape representations such as the use of relative coordinates or Fourier descriptors.

3.4.1 Learning Invariance

The way we introduced the shape prior in the previous sections fundamentally differs from many other approaches to introduce prior knowledge in computer vision tasks. Namely we pursue the paradigm of *learning from examples*. So we do not specify the knowledge about hands by a set of fixed rules such as "A hand consists of one big blob for the palm and five elongated blobs for the fingers...". Instead we present the machine a set of training images containing hand shapes, and let the machine "learn" by itself the rules that define the concept "hand". The advantages of machines which can learn in such a way are obvious: Not only are fixed rules difficult to define for more complicated recognition tasks, but also a machine which can learn in an unsupervised manner will on the long run advance much faster. Moreover, this concept of learning from examples is much closer to the way humans acquire knowledge.

The problem of integrating invariance can be tackled in a similar way. Assume that we are given a set of training shapes $\{z_i\}$, such that the correspondences between pairs of control points are correct. Upon acquisition of the shape vector, this set contains information about the location, rotation and scale of each shape. Previously we discarded this information by aligning the shape vectors with respect to similarity transformations. In the following we will analyze to what extent one can instead retain this information and learn the respective invariance in the process described in Section 3.3.

Assume that we store the center position for all shapes, then align them with respect to similarity transformations as done in Section 3.1.4, and finally add the respective center locations again. A Gaussian distribution estimated from this data will encode the translatory information. In fact, if the training shapes are sufficiently distributed over the image plane, then the deformation modes which correspond to translation in x- and y-direction will be favored by the Gaussian model of Section 3.3.

To illustrate this, we will assume for simplicity that all training shapes $\{z_i\}_{i=1,\dots,m}$ have been centered and aligned. Two additional training shapes are generated by translating the mean shape $\bar{z}$ given in equation (3.12) by a fixed amount in both directions along the x-axis:

$$z_{m+1} = \bar{z} + \gamma\, d_x, \quad z_{m+2} = \bar{z} - \gamma\, d_x, \quad \text{with } \gamma > 0,$$

where the vector $d_x = (1, 0, 1, 0, \dots)^t / N$ denotes the normalized translation in x-direction. For this artificially enlarged training set, the sample mean coincides with the old one, whereas the new sample covariance matrix $\hat{\Sigma}$ is given in terms

of the original one Σ by:

$$\hat{\Sigma} = \frac{1}{m+1} \sum_{i=1}^{m+2} (z_i - \bar{z})(z_i - \bar{z})^t = \frac{m-1}{m+1} \Sigma + \frac{2}{m+1} \gamma^2 \, d_x d_x^t.$$

Since the initial set of training shapes was centered, we have the orthogonality:

$$z_i^t d_x = 0 \quad \forall i = 1, \ldots, m, \quad \text{and} \;\; \Sigma \, d_x = 0.$$

Therefore we obtain:

$$\hat{\Sigma} \, d_x = \frac{2}{m+1} \; \gamma^2 \; d_x.$$

This means that translation along the x-direction appears as one of the eigenmodes of the new covariance matrix $\hat{\Sigma}$. The further the translation γ, the larger the corresponding eigenvalue and the more such deformations are favored by the shape energy (3.17). Note that the other eigenmodes e_i of Σ which represent the possible variations of shape are not affected, since they are by construction orthogonal to d_x. In the same way, translation d_y in y-direction can be "learnt".

In precise terminology, the resulting shape energy is *not invariant* with respect to translation, but rather *robust* to translation. In practice, however, the effect is almost identical if the translation γ in the learning step is sufficiently large. Moreover, such a *robustness* to similarity transformations is quite common for the human visual system, where numerous examples indicate that certain objects (such as faces) are reliably recognized only if the rotation from the default position does not exceed a certain amount. The further the rotation, the smaller the recognition rate. Translation learning within the Gaussian model will favor a certain position in the image plane (the location of the mean shape) and penalize deviations from this location.

If the initial shape set is aligned but not centered, then the eigenmodes of the covariance matrix will in general be linear combinations of the translatory degrees of freedom d_x and d_y and the other modes $\{e_i\}$ of shape deformation. We will not go into detail about this.

This method of translation learning relies on the fact that there exist vectors d_x and d_y associated with translatory motion which are independent of the particular translated shape and which are orthogonal to the shape deformation modes. Unfortunately, this is not the case for rotation and scaling. Therefore the latter two invariances cannot be learnt in the Gaussian model. The eigenvectors which model the rotation and scaling cannot be separated from the shape z. The concepts of scaling and rotation of *one* shape cannot be extended to other shapes.

The following example will illustrate in what way rotation and scaling are always associated with a particular shape. We denote the rotation matrix for a $2N$-dimensional control point polygon by

$$R_\theta = \mathrm{I}_N \otimes \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3.19}$$

Assume that the training set contains only one shape in all possible rotations $\{R_\theta\, z\}$, such that all rotation angles $\theta \in [0, 2\pi]$ appear equally often. Furthermore, assume the shape z to be normalized: $z^t z = 1$. We will show that all eigenmodes of the covariance matrix are given by scaled and rotated versions of the shape z. The mean shape is given by

$$\bar{z} = \frac{1}{2\pi} \int_0^{2\pi} R_\theta\, z\; d\theta,$$

and the covariance matrix is given by

$$\Sigma = \frac{1}{2\pi} \int_0^{2\pi} R_\theta\, z\, (R_\theta\, z)^t\, d\theta.$$

This matrix is symmetric and positive semidefinite. All eigenvalues are real and nonnegative. Their sum is given by:

$$\mathrm{tr}\,[\Sigma] = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{tr}\left[R_\theta\, z\, (R_\theta\, z)^t\right] d\theta = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{tr}\left[z^t z\right] d\theta = 1.$$

The corresponding eigenvalue system is given by $\{\frac{1}{2}, \frac{1}{2}\}$, and the eigenvectors are given by $R_\mu z$ for any angle $\mu \in [0, 2\pi]$:

$$\Sigma R_\mu z = \frac{1}{2\pi} \int_0^{2\pi} R_\theta z z^t R_{\mu-\theta} z d\theta = \left[\frac{1}{2\pi} \int_0^{2\pi} R_\theta \cos(\mu - \theta) d\theta\right] z = \frac{1}{2} R_\mu z.$$

Obviously the eigenvectors span the same 2-dimensional linear subspace if the training set contains only three linearly independent rotations of the input shape z. Yet scaling and rotation are not learnt: The eigenvectors $R_\mu z$ associated with scaling and rotation cannot be separated from the particular shape z! Therefore the concepts of rotation and scaling cannot be generalized to other shapes, as was the case for translation.

As a possible remedy, one could switch from a control point representation in Cartesian coordinates x and y to a representation in cylindrical coordinates $r = \sqrt{x^2 + y^2}$ and $\phi = \arctan\left(\frac{y}{x}\right)$:

$$z = (r_1, \phi_1, \ldots, r_N, \phi_N).$$

In principle this would permit the learning of rotation and scaling, since in this representation both of these operations correspond to deformations along the constant vectors $(1, 0, 1, 0, \ldots)^t$ and $(0, 1, 0, 1, \ldots)^t$. However, first of all it is difficult to deal with the inherent ambiguities associated with the periodicity of the angle coordinate. And secondly, this solution would only shift the problem, since in cylindrical coordinates translation can no longer be learnt.

3.4.2 Variational Integration of Invariance

Rather than trying to *learn* the invariances from the example shapes in the linear model of shape statistics, one can *construct* them explicitly. In the following we shall demonstrate how this can be done for the group of similarity transformations. However, other invariances — such as the more general invariance to affine transformations — can be incorporated in the same manner.

Let $\{z_i\}$ be a set of training shapes, aligned as detailed in Section 3.1.4, and let $E(z)$ be the energy (3.17) corresponding to a Gaussian model of shape probability. There are several ways to incorporate invariance under a certain transformation group into (3.17). For example, one can simply integrate or minimize over this group:

$$E_{int}(z) = \int E\big(sR_\theta(z-t)\big)\,ds\,d\theta\,dt, \tag{3.20}$$

$$E_{min}(z) = \min_{s,\theta,t} E\big(sR_\theta(z-t)\big). \tag{3.21}$$

However, both of these approaches have certain drawbacks. For translation and scaling, the integration range has to be restricted for the integral to be well defined. Moreover, the integration produces an average of energy (3.17) over all possible transformations. Although the resulting measure is by construction invariant to these transformations, this averaging process will generally not produce a sensible measure of shape dissimilarity: The value of energy (3.17) evaluated for an *incorrectly* translated or rotated shape vector should not affect the final measure, yet it does in the integration in equation (3.20), left side.

The minimization in (3.21) is therefore a much better solution. However, as for the integration, a closed-form solution for this minimization appears infeasible. A simplification of this minimization is the following approach which *can* be solved analytically.

Since the training shapes were aligned to their mean shape $\bar{z}$ with respect to translation, rotation and scaling and then normalized to unit size (cf. Section 3.1.4), the same should be done to the argument z before applying energy (3.17). This is detailed in the following.

We eliminate the translation by centering the control point vector (3.2):

$$z_c = \left(I_{2N} - \frac{1}{N}T\right) z, \tag{3.22}$$

where I_{2N} denotes the unit matrix of size $2N$, N is the number of control points, and the $2N \times 2N$-matrix T is given by:

$$T = \begin{pmatrix} 1 & 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & 1 & \cdots \\ 1 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Next, we eliminate rotation and scaling by aligning with respect to the mean of the training data. The final shape energy expressed in terms of the original

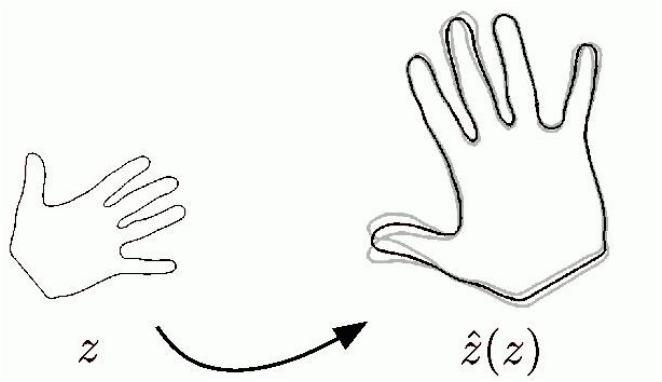


Figure 3.6: For a given shape vector z, a similarity invariant shape energy is obtained by applying the statistical energy (3.17) to the vector $\hat{z}$ determined by registration with the training set (depicted in gray), see equation (3.23).

energy E in (3.17) is given by:

$$\boxed{E_{shape}(z) = E(\hat{z}), \quad \text{with } \hat{z} = \frac{R_\theta\, z_c}{|R_\theta\, z_c|}\,,} \tag{3.23}$$

where θ denotes the angle corresponding to the optimal rotation of the centered control point polygon z_c with respect to the mean shape $\bar{z}$. Figure 3.6 shows a schematic drawing of this intrinsic alignment. In Section 3.1.4, we discussed the alignment in complex notation. Conversion of (3.10) to the equivalent representation in real coordinates gives the formula:

$$\hat{z} = \frac{M\, z_c}{|M\, z_c|}, \quad \text{with } M = I_N \otimes \begin{pmatrix} \bar{z}^t\, z_c & -\bar{z} \times z_c \\ \bar{z} \times z_c & \bar{z}^t\, z_c \end{pmatrix}, \tag{3.24}$$

where $\otimes$ denotes the Kronecker product and $\bar{z} \times z_c := \bar{z}^t R_{\pi/2}\, z_c$.

Given an initial shape z, one can maximize the similarity with respect to the set of training shapes by performing a gradient descent on the final shape energy (3.23). In order to determine the gradient of (3.23), we will denote the differentiation of vector-valued functions $f : \mathbb{R}^n \to \mathbb{R}^m$ by:

$$\frac{df}{dx} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

Applying the chain rule for differentiation to (3.23), one obtains the following gradient descent equation:

$$\frac{dz}{dt} = -\frac{dE_{shape}(z)}{dz} = -\frac{dE(\hat{z})}{d\hat{z}} \cdot \frac{d\hat{z}}{dz_c} \cdot \frac{dz_c}{dz}. \tag{3.25}$$

The three terms in this product can be interpreted as follows:

- The first term is the gradient of the original energy evaluated for the aligned shape $\hat{z}$. It contains the shape information extracted from the training set. For the linear model (3.17) it is given by:

$$\frac{dE(\hat{z})}{d\hat{z}} = \left(\Sigma_{\perp}^{-1}\left(\hat{z} - \bar{z}\right)\right)^t.$$

  It causes a relaxation of the aligned shape $\hat{z}$ towards the mean shape $\bar{z}$, weighted by the inverse of the regularized covariance matrix. This weighting causes unfamiliar deformations from the mean to decay faster.

- The second term in the product of (3.25) takes into account the influence of changes in the centered shape z_c onto the aligned shape $\hat{z}$. In matrix notation it is given by:

$$\frac{d\hat{z}}{dz_c} = \frac{M' z_c + M}{||Mz_c||} - \frac{(Mz_c)(Mz_c)^t(M' z_c + M)}{||Mz_c||^3}, \tag{3.26}$$

  where M is the matrix defined in (3.24) and M' denotes the tensor of rank 3 given by:

$$M' = \frac{dM}{dz_c}.$$

  Using the real notation (3.2) for the control point vectors and the definition of M in (3.24), the entries of this constant sparse tensor are given by:

$$M'_{ijk} = \frac{dM_{ik}}{d(z_c)_j} = \begin{cases} \bar{z}_j, & i = k \\ \bar{z}_{j+1}, & i = k+1,\ i \text{ even},\ j \text{ odd} \\ -\bar{z}_{j-1}, & i = k+1,\ i \text{ even},\ j \text{ even} \\ -\bar{z}_{j+1}, & i = k-1,\ i \text{ odd},\ j \text{ odd} \\ \bar{z}_{j-1}, & i = k-1,\ i \text{ odd},\ j \text{ even} \\ 0, & \text{otherwise} \end{cases}.$$

- The third term in the product of (3.25) accounts for the change of the centered shape z_c with the input shape z. According to definition (3.22), it is given by:

$$\frac{dz_c}{dz} = \left(I_{2N} - \frac{1}{N}T\right).$$

  This term centers the energy gradient, as a direct consequence of the translation invariance of the shape energy. This means that the force which minimizes the shape energy has no influence on the translation of the contour. Similarly, rotation and scaling of the shape are not affected by the shape optimization, due to the term (3.26) in the evolution equation (3.25).

Note that the final shape energy (3.23) is *by definition* invariant with respect to similarity transformations of the contour. In particular, this variational integration is entirely parameter-free!

This closed-form solution to incorporate invariance can be extended to the more general group of *affine transformations*, since there exist closed-form solutions for the alignment of two polygons with respect to affine transformations — see [190]. However, we will not pursue this idea in the present work.

3.4.3 Alternative Approaches to Invariance

Optimization of Explicit Pose Parameters

A common alternative to treat similarity transformations of the shape (cf. [47, 93, 106, 17, 120, 36]) is to introduce explicit *pose parameters* to account for the degrees of freedom associated with translation, scaling and rotation. Then a joint optimization with respect to both the shape parameters corresponding to shape deformation modes and the pose parameters is performed. This was done by deterministic approaches [47, 36], by stochastic approaches such as genetic algorithms [93], or by simulated annealing on the basis of the Gibbs sampler [106].

Since the gradient descent presented in Section 3.4.2 constitutes a deterministic approach, we will for comparison present the corresponding approach with explicit pose parameters, derived in analogy to the one presented in [36].

The shape energy E_{shape} is defined in terms of the quadratic energy E in (3.17) by:

$$E_{shape}(z, s, x, y, \theta) = E(\hat{z}), \quad \text{with } \hat{z} = s\, R_\theta\, (z + x\, d_x + y\, d_y),$$

where the pose parameters s, x, y, θ stand for scaling, translation and rotation, respectively, and with the vectors $d_x = (1, 0, 1, \ldots, 0)^t$ and $d_y = (0, 1, 0, \ldots, 1)^t$ again denoting translation of the shape in x- and y-direction.

These parameters are assigned appropriate initial values, and shape and pose are both optimized by iterating the respective gradient descent equations for translation

$$\frac{dx}{dt} = -\frac{dE_{shape}}{dx} = -\frac{dE(\hat{z})}{d\hat{z}}\frac{d\hat{z}}{dx} = -\frac{dE(\hat{z})}{d\hat{z}}\, s\, R_\theta\, d_x, \tag{3.27}$$

for scaling

$$\frac{ds}{dt} = -\frac{dE_{shape}}{ds} = -\frac{dE(\hat{z})}{d\hat{z}}\frac{d\hat{z}}{ds} = -\frac{dE(\hat{z})}{d\hat{z}} R_\theta\, (z + x\, d_x + y\, d_y), \tag{3.28}$$

for rotation

$$\frac{d\theta}{dt} = -\frac{dE_{shape}}{d\theta} = -\frac{dE(\hat{z})}{d\hat{z}}\frac{d\hat{z}}{d\theta} = -\frac{dE(\hat{z})}{d\hat{z}}\, s\, R'_\theta\, (z + x\, d_x + y\, d_y), \tag{3.29}$$

and for shape

$$\frac{dz}{dt} = -\frac{dE_{shape}}{dz} = -\frac{dE(\hat{z})}{d\hat{z}}\frac{d\hat{z}}{dz} = -\frac{dE(\hat{z})}{d\hat{z}} s R_\theta. \tag{3.30}$$

The matrix R'_θ in (3.29) denotes the derivative of the matrix R_θ defined in (3.19):

$$R'_\theta = \frac{dR_\theta}{d\theta} = \mathrm{I}_N \otimes \begin{pmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix}.$$

As can be seen in equations (3.27), (3.28), (3.29) and (3.30), the evolution of the respective parameters is simply given by appropriate projections of the gradient of the original energy (3.17) evaluated at the transformed shape $\hat{z}$.

Compared to the closed-form solution proposed in Section 3.4.2, this latter approach has several drawbacks:

- It mixes the degrees of freedom associated with shape deformation and those associated with pose. This can be improved by restricting the shape deformation to the shape eigenmodes $\{e_i\}$ of the covariance matrix of the training set, as detailed in Section 3.2:

  $$E_{shape}(\alpha, s, x, y, \theta) = E(\hat{z}), \quad \text{with } \hat{z} = sR_\theta(\bar{z} + \sum_i \alpha_i e_i + x\, d_x + y\, d_y).$$

  The evolution equation (3.30) is then replaced by corresponding evolution equations for the shape parameters $\alpha = \{\alpha_i\}$:

  $$\frac{d\alpha_i}{dt} = -\frac{dE_{shape}}{d\alpha_i} = -\frac{dE(\hat{z})}{d\hat{z}}\frac{d\hat{z}}{d\alpha_i} = -\frac{dE(\hat{z})}{d\hat{z}} s\, R_\theta\, e_i.$$

  This separates shape and translation, because the shape eigenmodes are orthogonal to the translation vectors d_x and d_y. However, it does not fully separate the deformation modes e_i from rotation and scale. In contrast, in the approach of Section 3.4.2, we first enforce the optimal similarity transformation and then perform a gradient descent. This induces a *clear separation of shape and pose* in the sense that only the deformation which cannot be "explained" by similarity transformations will be associated with shape deformation.

- Compared to the closed-form solution of Section 3.4.2, we need to find appropriate parameters associated with each pose parameter in order to balance the different gradient descent equations. In numerical implementations this may be tedious. Moreover, too large step sizes may introduce numerical instabilities.

- Additional local minima may be introduced by the pose parameters. In a given application, this may prevent the convergence of the contour towards the desired segmentation.

On several segmentation tasks we were able to confirm these effects by comparing the two approaches — see Section 3.6.5.

Alternative Shape Representations

Another approach to introduce invariance is to revert to contour representations which are intrinsically invariant with respect to certain transformations. From the large number of possible representations, we will briefly discuss two of these, namely the use of relative coordinates and the use of Fourier descriptors.

A straight-forward extension of the representation used in the previous sections is to employ *relative coordinates* in order to enforce translational invariance. These can be defined e.g. relatively to a fixed control point or relatively to the center point. To additionally enforce rotation and scale invariance, a further extension is to describe the connecting line segments of the control point polygon in terms of their lengths $\{l_i\}$ and their angles $\{\phi_i\}$, both measured relatively to a reference line such as the first segment or the medial axis. This representation is invariant to similarity transformations of the contour. However, the invariance comes at a certain cost: Firstly, the intrinsic ambiguities of the angle representation are difficult to cope with when performing shape statistics. Secondly, the relative coordinates produce undesired dependencies: If measured relatively to the first segment, a perturbation of the first control point will affect all coordinates, and if measured relatively to the medial axis, all coordinates will show a discontinuous behavior for shape deformations which modify the medial axis — a circle would for example define an instable control point configuration.

Rather than working in the spatial domain, one can revert to the Fourier domain [201, 86, 146, 114, 170]. *Fourier descriptors* are obtained by applying the discrete Fourier transform to the contour. By construction, the Fourier descriptors are invariant with respect to translation. Appropriate modifications permit to incorporate similarity invariance [28] and even affine invariance [8]. Moreover, Fourier descriptors present a multiresolution description of shape.

Yet, in our context, the use of Fourier descriptors has two disadvantages: Firstly, it introduces additional computational complexity. Since the image information driving the contour evolution is determined in the spatial domain, working with Fourier descriptors would require to constantly transform between the two domains. Secondly, by construction the Fourier descriptors represent spatially global deformation modes. With a limited number of frequency components one cannot model spatially very localized changes of the contour — such as the motion of a single finger when the rest of the hand is stable.

Similar arguments hold for *moment invariants* [97, 182]. A spatially localized version of the Fourier descriptors is given by *wavelet descriptors* [193, 148, 38]. A detailed analysis of these approaches is beyond the scope of this work.

3.5 Linear Shape Statistics in Segmentation

In this section, we combine the linear shape prior introduced in Section 3.3 with the diffusion snakes introduced in Sections 2.4 and 2.5. The result is a variational approach to segmentation, which incorporates both information from the input image and statistically encoded prior knowledge about the shape

of the segmented object. We propose to minimize the total energy given by:

$$E(z) = E_{image}\,(u, C_z) \;+\; \alpha\, E_{shape}(z), \tag{3.31}$$

where the image energy E_{image} is either the **DS** in (2.20) or its simplified version **SDS** in (2.24).

Invariance of the shape prior will be introduced either in terms of learnt translation invariance as proposed in Section 3.4.1, or in terms of the closed-form variational integration of similarity invariance as proposed in Section 3.4.2.

In the case of translation learning, the two translational degrees of freedom appear as two eigenvectors of the covariance matrix and the shape energy E_{shape} is simply given by the quadratic energy (3.17). The gradient descent equations for the control point (x_m, y_m) are then given by:

$$\begin{aligned}
\frac{dx_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \left[\, \left(e^+(s_i,t) - e^-(s_i,t)\right) \boldsymbol{n}_x(s_i,t) \;+\; \nu(x_{i-1} - 2x_i + x_{i+1}) \right] \\
&\quad - \alpha \, \left[\Sigma_\perp^{-1}\, (z - \bar{z})\right]_{2m-1}, \\
\frac{dy_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \left[\, \left(e^+(s_i,t) - e^-(s_i,t)\right) \boldsymbol{n}_y(s_i,t) \;+\; \nu(y_{i-1} - 2y_i + y_{i+1}) \right] \\
&\quad - \alpha \, \left[\Sigma_\perp^{-1}\, (z - \bar{z})\right]_{2m}.
\end{aligned} \tag{3.32}$$

This simply extends the contour evolution equations in (2.30) by the last term which maximizes the similarity of the evolving contour with respect to the set of training shapes. The three terms in the respective equations in (3.32) can be interpreted as follows:

- The first term forces the contour towards the object boundaries, maximizing a homogeneity criterion in the adjoining regions which compete in terms of their energy densities e^+ and e^-. Depending on the model — piecewise smooth or piecewise constant grey value — they are given by (2.26) and (2.27), respectively.

- The second term enforces an equidistant spacing of control points by minimizing the length measure (2.21).

- The last term causes a relaxation towards the most probable shape, by minimizing the shape energy. The indices $2m - 1$ and $2m$ are simply associated with the x- and y-coordinates of the m-th control point in the notation of (3.2). Note that the relaxation towards the most probable shape is weighted by the inverse of the modified covariance matrix, such that less familiar shape deformations will decay faster. This interesting property arises automatically due to the proposed variational integration of the prior.

For full similarity invariance of the shape prior, the energy E_{shape} is given by (3.23). Consequently, the gradient of the shape energy, given by the last term in the evolution equations (3.32), has to be modified as detailed in Section 3.4.2.

3.6 Numerical Results

In this section, we will present numerical results of the segmentation approach which combines the linear shape prior with the diffusion snakes. Depending on the context, we introduce invariance either in terms of learnt translation invariance as proposed in Section 3.4.1 or full similarity invariance as proposed in Section 3.4.2. For all results we used a fixed number of $N = 100$ control points, as this gives sufficient resolution for the objects which are to be segmented. The specific number of control points is not crucial. It should, however, be constant for the shape statistics to be well defined. If not specified otherwise, we generally show the input image and the contour C which minimizes the total energy (3.31).

3.6.1 Image-driven versus Knowledge-driven Segmentation

The total energy (3.31) is a weighted sum of an image energy and a shape energy. Minimizing the image energy forces the contour towards the boundaries of the object as suggested by the grey value information. Minimizing the shape energy forces the contour towards the boundaries of the object as indicated by

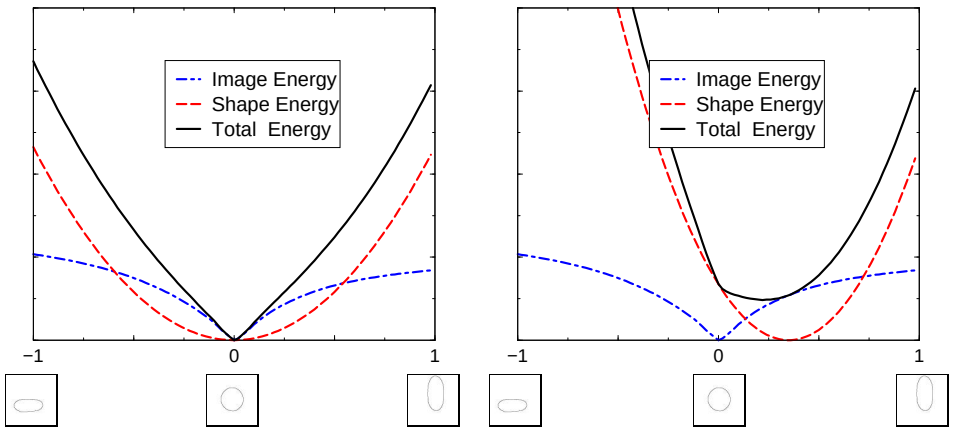


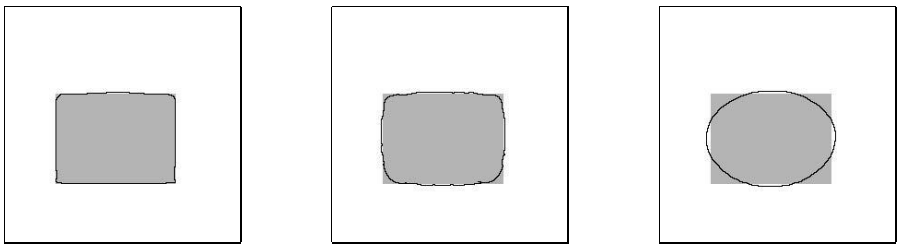
Figure 3.7: Energy plots. For two different training sets, both plots show the Mumford-Shah image energy E_{image}, the shape energy E_{shape} and the total energy for a fixed input image as a function of different contours. For the parameter values $-1, 0, 1$, the three respective contours are shown below. The object in the image corresponds to the contour in the middle. For the training set on the left, the mean (i.e. most probable) shape and the input object are the same, such that the total energy is simply a convex version of the image energy having the same minimum. For the training set on the right, input object and mean shape are not the same, so that the position of the minimum is shifted by the prior. Minimization of the total energy produces a "compromise" between image and shape information. Note that in both cases, the total energy is convex while the image energy by itself is not.

the statistically learnt object notion. If the minima of both energies are the same, then adding the shape energy tends to "convexify" the total energy, since most contour deformation modes are effectively suppressed by the prior. If there is a discrepancy between the input intensity information and the learnt object notion, then minimization of the total energy will produce a segmentation which is a weighted compromise between that indicated by the intensity information and that favored by the shape prior. These effects are explained on the basis of energy plots in Figure 3.7.

Increasing the weighting parameter α in the total energy (3.31) allows to continuously shift from a purely image-based segmentation to one which mostly relies on the learnt shape information. The following results will demonstrate this aspect.

Ellipse versus Rectangle

Figure 3.8 gives a simple demonstration of the effect of the shape prior on the segmentation result: Without prior knowledge, the segmented image coincides with the input image (left contour), whereas the results for larger knowledge weight α (middle and right contour) are shifted towards the prior information, which encodes a set of six ellipses. We used the diffusion snake model (2.20) for the image energy and learnt translation invariance for the shape prior.



No prior Weak prior Strong prior

Figure 3.8: Rectangle with prior favoring ellipses. The knowledge energy was calculated on a set of six ellipses. The input image (grey square) and the final contour (black) are shown for increasing values of the knowledge weight α.

Parsing an Image into its Constituent Components

The second example shows that the effect of the shape prior can be exploited in order to deal with occlusions of the object of interest. Figure 3.9 shows an input image which can be interpreted as an ellipse covered by a bar. The contours correspond to the respective minima of the total energy (3.31) for the diffusion model (2.20) and two different priors. For the top row, we used a prior constructed on a set of six ellipses (which were not aligned), with translation invariance introduced as detailed in Section 3.4.1. For the bottom row, we constructed a prior from a set of four vertical bars. The final segmentations show

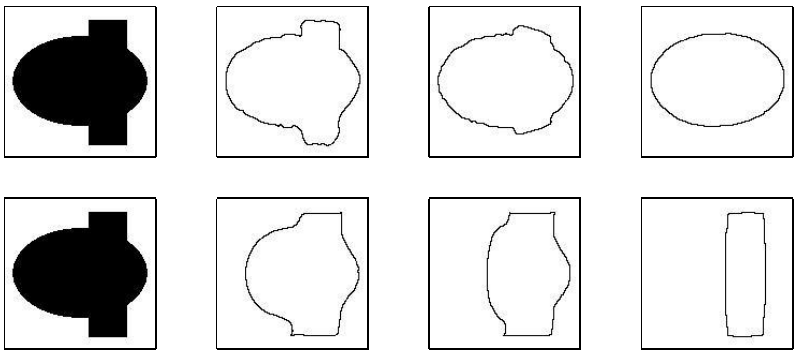
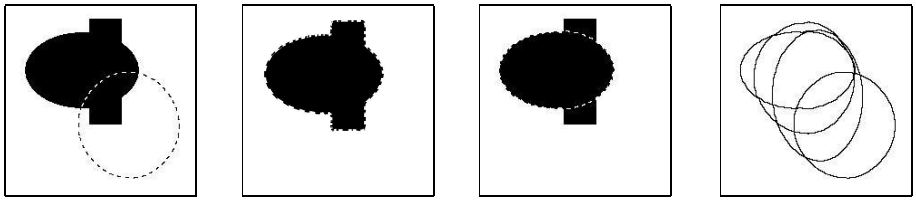


Figure 3.9: Ellipse covered by a bar. Input image (left) and segmenting contours for a prior favoring ellipses (top row) and a prior favoring bars (bottom row). With increasing knowledge strength α (from left to right) unfamiliar shape deformations are suppressed. The input image is parsed into its constituent components by applying different priors.

that for increasing weight α of the prior, the segmentation process continuously ignores shape deformations that are less probable according to the respective shape statistics. Note, however, that for both priors the resulting contour does *not* simply correspond to the mean shape, i.e. the most probable shape of the respective model. Even for large values of α (Figure 3.9, right side), the segmentation process still incorporates evidence given by the input image. In this example, we can actually parse the image into its constituent components, by specifying different objects of interest with different priors.



Initial contour | Segmentation without prior | Segmentation with prior | Evolution with prior

Figure 3.10: Segmentation results for the **SDS** with a prior favoring ellipse-like shapes. Some intermediate contours (right) indicate how the contour evolution is restricted to the submanifold of familiar shapes.

Reducing the Effective Dimension of the Search Space

The Gaussian shape prior effectively restricts the evolving contour to the linear subspace of familiar contours spanned by the first few eigenmodes of the covariance matrix.[3] For the example from Figure 3.9, this is visualized in Figure 3.10. It shows the initial contour, the segmentation without prior and the

[3]Due to the regularization (3.15) of the covariance matrix, the restriction to the subspace spanned by the principal components is incorporated as a *soft* constraint — see Section 3.3.2.

segmentation with prior. Some intermediate contours, shown in the last image, indicate how the evolving contour is restricted to ellipse-like shapes during the gradient descent minimization.

3.6.2 Translation Learning

In the examples in Figures 3.9 and 3.10, invariance of the shape prior with respect to translation was "learnt" as detailed in Section 3.4.1. The training shapes were not aligned at all, their locations being only more or less concentric. As discussed in Section 3.4.1, this already induces some robustness to translation. By additionally translating the mean shape along the two orthogonal spatial directions and updating the covariance matrix accordingly, the translatory degrees of freedom will become energetically more favorable and translatory motion is facilitated. The effect of enabling translatory motion on the knowledge-driven segmentation process is demonstrated in Figure 3.11. Without the translation learning, the center of the final contour has shifted only marginally from its initial location. After translation learning, translatory motion of the contour is no longer suppressed by the prior.

As discussed in Section 3.4.1, learning invariance cannot be extended to rotation and scaling. Moreover, the shape prior after translation learning is not *invariant* but rather *robust* against translation. Because of these limitations, we will not investigate the issue of learning invariances any further. Instead, in all following examples we will only employ the closed-form variational integration of translation or similarity invariance as proposed in Section 3.4.2.

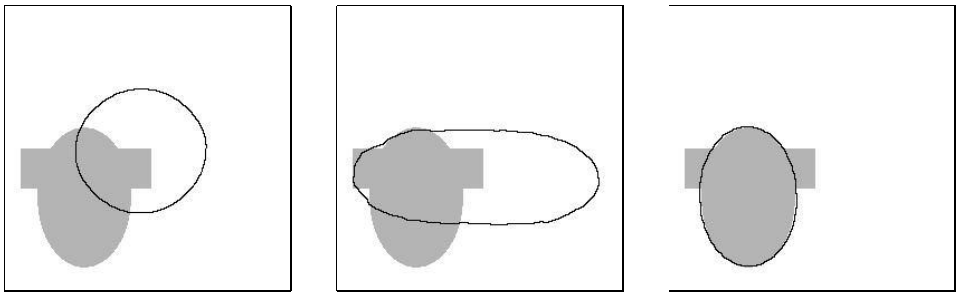
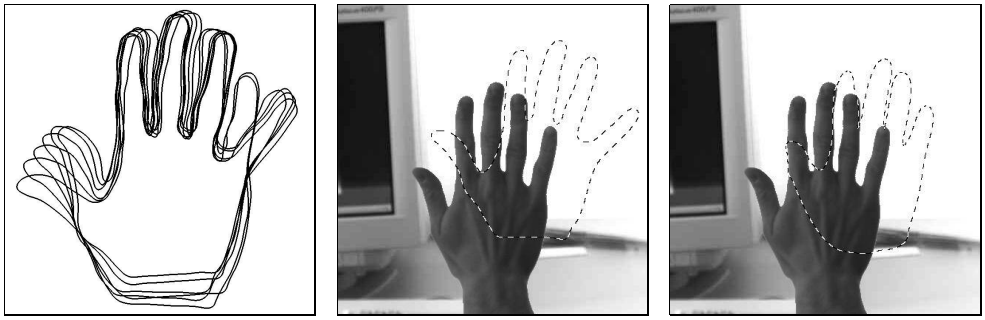


Figure 3.11: Translation learning. The input image (grey) is an ellipse occluded by a bar. The **left image** shows the initial contour. The shape energy was constructed on a set of 6 (more or less concentric) horizontal and vertical ellipses, which effectively restricts contour variation to elliptical shapes, the translatory degrees of freedom being essentially suppressed. The **middle image** shows the final contour *without* translation learning: The center of the ellipse has essentially not moved. The **right image** shows the final contour *with* translation learning: The shape energy is much less sensitive to translation of the shape, and the ellipse that best describes the input image is found. Due to the statistical a priori-knowledge, the bar is ignored by the diffusion snake despite the prominent signal transitions at its boundary.



Aligned training set | Initial contour | Intermediate 1

Intermediate 2 | Final segmentation | Result *without* prior

Figure 3.12: Segmentation with statistical prior. Aligned training contours and contour evolution from initial to final step. For comparison, the corresponding segmentation without the prior is shown on the bottom right. The second intermediate step indicates that the embedding of the shape prior as a *soft* constraint permits some shape deformation outside the subspace spanned by the training shapes.

3.6.3 Coping with Clutter

In the following, we denote as *clutter* all noise which is spatially structured (i.e. not fully random). Clutter in the background may be strongly misleading for the Mumford-Shah based segmentation approach, since it violates the hypothesis of constant or smooth grey value information. The statistical prior permits the segmenting contour to "ignore" such misleading information by restricting it to the subspace of familiar shapes.

To demonstrate this property, we go back to the example image of Figure 2.6, for which — although it did not contain a lot of clutter — we did not obtain the desired segmentation without a shape prior. In order to include prior information, we constructed a shape energy upon a set of six binarized hand images as explained in Sections 3.1.3 and 3.1.4. The hand in Figure 3.12 was *not* part of the training set. The aligned training contours are shown in Figure 3.12, top left. From the invariances suggested in Section 3.4.2 we only included translation. The training shapes all had the same rotation and scale as the object in the image. Results which also include scale and rotation invariance

will be shown separately later on.

For the same input image and the same initial contour as in the example of Figure 2.6, we then performed a gradient descent on the full energy (3.31) for the **DS** (2.20)[4]. Figure 3.12 shows three steps in the contour evolution from the initialization to the final segmentation. For a comparison, the image on the bottom right shows the corresponding segmentation obtained without the shape prior.

The statistical prior effectively restricts the contour deformations to the subspace of learnt deformations. However, due to the embedding of the shape probability into the full space of possible deformations, as explained in Section 3.3, some deformation outside this subspace is still feasible — as can be seen in the intermediate steps in Figure 3.12. This *flexibility* turns out to strongly improve the ability of the system to evade incorrect local segmentations. The final segmentation is cut at the wrist, since the training shapes were all cropped there for simplicity.

The question of which value to assign to the length-governing parameter ν in equations (2.20) and (2.24), discussed in Section 2.6, becomes obsolete: The effective restriction of shape deformations imposed by the prior allows to drop the additional length minimization term. However, for the purpose of analyzing the effect of the prior we kept the value of ν constant for the segmentations with and without prior.

The scene in Figure 3.12 contains little clutter. Therefore segmentation results are rather good even in the case when no prior knowledge is included. Once the amount of clutter is increased, this changes. Therefore, we go back to the example in Figure 2.7, which shows a hand in front of a strongly cluttered background. Note that the grey value of the background is approximately the same as that of the object of interest. Without the statistical prior, none of the segmentation approaches compared in Section 2.6 is able to extract the object of interest.[5] In the example of Figure 2.7, the hypothesis of constant or smooth grey value is not valid for the background, such that the Mumford-Shah based segmentation *without* shape prior lead to unsatisfactory results.

As in the previous example, we now include the shape prior and perform a gradient descent on the total energy (3.31) to obtain the segmentation shown in Figure 3.13, top row, for the case of the diffusion snake functional with and without statistical prior. Again, the shape in the image was not part of the training set.

The final segmentation produced with the statistical prior is the desired one. Small discrepancies between the object boundary and the final contour in the area between the fingers are probably due to the fact that the shape prior does not fully suppress some shape variability in that area. This could be improved

[4]Segmentation results of equal quality as in Figure 3.12 were obtained by including a statistical shape prior in the **SDS** (2.24).

[5]Since there exists a vast amount of very different segmentation approaches, there may be some which produce an adequate segmentation of the hand in the image of Figure 2.7. Yet, for *any* segmentation approach which *only* takes into account the information contained in the grey values, one can always find an example image for which the desired segmentation is not obtained.

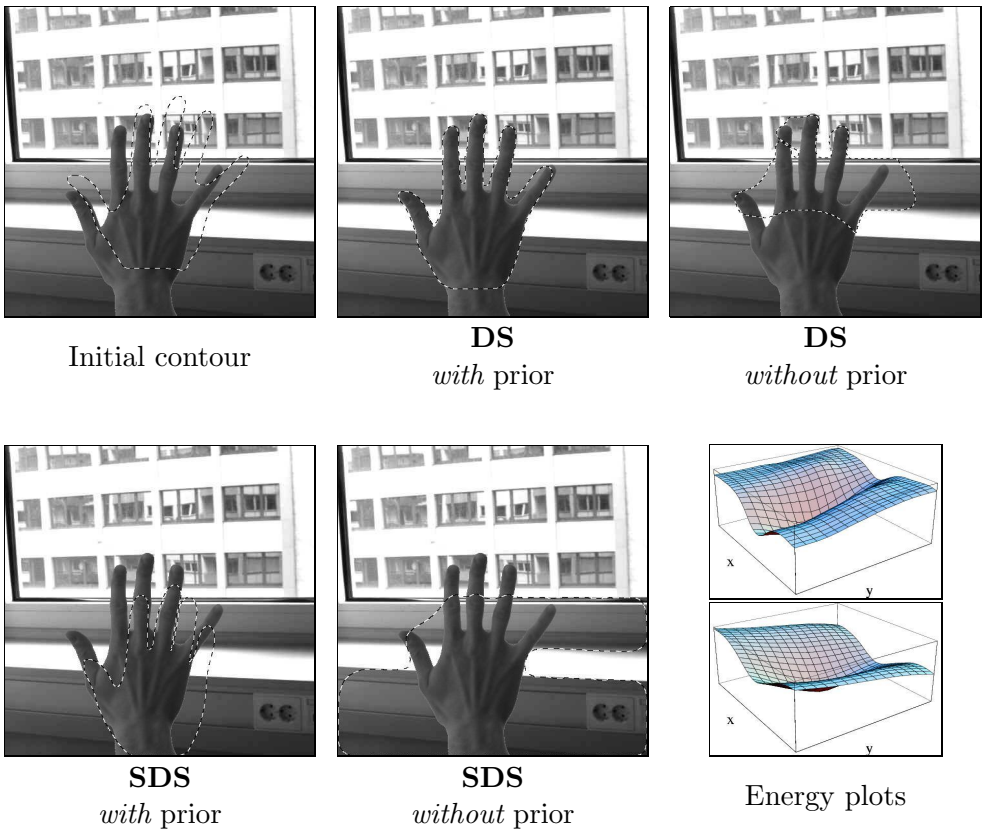


Figure 3.13: Object in strongly cluttered environment. Results of segmentation with and without shape prior for the diffusion snake (2.20) and its cartoon limit (2.24). Note that the cartoon model (bottom row) does not produce the desired segmentation even with the prior. The bottom right image shows associated energy plots for the **DS** (top) and the **SDS** (bottom) — see text.

with a more elaborate alignment of the training shapes during shape learning. However, we decided to avoid any shape learning that involves the calculation of landmarks or any manual interaction such as the labeling of correspondences.

The segmentation obtained with statistical prior in the case of the cartoon model (**SDS**) was not successful, as can be seen in Figure 3.13, bottom row. The reason for this failure to capture the object of interest will be discussed next.

3.6.4 Comparing the Diffusion Snake and its Cartoon Limit

The full Mumford-Shah functional and its cartoon limit differ in their contour evolution equation in that the former collects grey value information from the area surrounding the contour by means of a diffusion process, whereas the latter does this by separately averaging over the areas adjacent to the respective contour point. The images in Figure 3.14 show the piecewise smooth approxi-

Original $\lambda = 10$ $\lambda = 20$ $\lambda = 40$ $\lambda = 1000$

Figure 3.14: From the diffusion snake to the cartoon limit. Original image f and diffused versions u for a fixed contour. With growing values of the smoothing parameter λ in (2.20), the amount of information about the local context is reduced. The contour is modeled by edgels "between pixels", such that all pixels belong to one of the adjacent regions and are therefore affected by the diffusion process.

mation u of a given input image which minimizes the diffusion snake functional (2.20) for a fixed contour C and increasing values of the smoothing parameter λ. These images demonstrate how the piecewise smooth approximation u converges to the piecewise constant approximation of the cartoon limit for $\lambda \to \infty$. This approximation u is the basis of the contour evolution — see equations (2.25) and (2.26). Therefore, the motion of the diffusion snake (**DS**) is affected mostly by the image information in the neighborhood of the respective contour point, the size of which is determined by the scale parameter λ. The cartoon snake (**SDS**), however, is equally affected by information in any part of the image. This explains the very different segmentation results obtained for the image in Figure 3.13, both with and without prior.

The segmentation obtained with the simplified diffusion snake (**SDS**) will be affected by grey value differences on a global scale. To analyze which effect this property has upon the energy landscape, we calculated the value of the diffusion snake functional and its cartoon limit for a fixed contour which we simply translated in x- and y-direction. This corresponds to a suppression of shape deformation. We used the same input image as in Figure 3.13. The contour was optimally placed upon the hand boundaries and then shifted in equidistant steps up to 30 pixels in each direction. The resulting energies are plotted in Figure 3.13, bottom right, as a function of the displacement from the optimal position. Note that the bottom of the input image corresponds to the top right side of the energy plots. Both energies show a minimum at the optimal position of the contour. However, the energy for the **SDS** (below) is strongly slanted towards the bottom of the image. This is caused by the global change in brightness of the input image from the top towards the bottom. It is in fact this global change in brightness which drives the contour to segment the entire bottom part of the image if no prior is given — see Figure 3.13, bottom row. Even in the case of added shape prior, the hand contour is pushed to the bottom of the image for the **SDS**. Stated in other words: The model of piecewise *smooth* grey value is more robust to grey value variations on a *global* scale than the model of piecewise *constant* grey value.

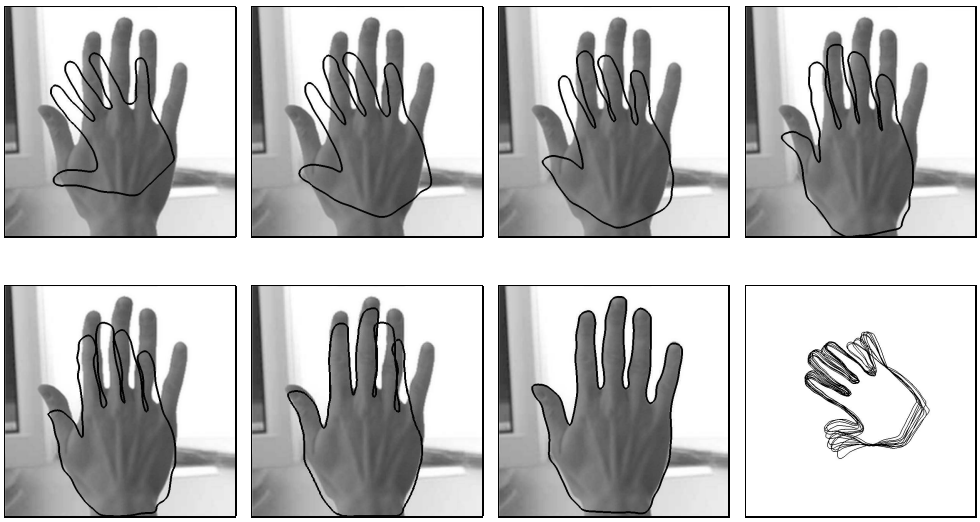


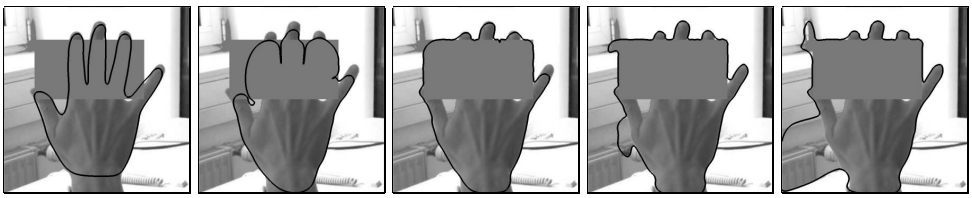
Figure 3.15: Invariance with respect to similarity transformation. Minimization by gradient descent from the initial contour (top left) to the final segmentation. Note that due to the closed-form solution (3.23), no additional parameters enter the minimization to account for scale, rotation and translation. Due to the intrinsic alignment of the evolving contour, the relative position, scale and rotation of the training set (bottom right) is of no importance to the knowledge-driven segmentation process.

3.6.5 Invariance to Similarity Transformations

By construction the shape energy (3.23) is invariant with respect to translation, rotation and scaling. Figure 3.15 shows a minimization by gradient descent from the initial contour (top left) to the final segmentation (bottom right), with a shape prior constructed from a set of 10 binarized hand images. During its evolution the contour is effectively restricted to the subspace of familiar contours, but translation, rotation and scaling are permitted.

The bottom right image in Figure 3.15 shows the training set of aligned hand shapes. The relative location, size and rotation of the training shapes is of no importance to the segmentation process, because the evolving contour is intrinsically aligned with the mean of the training shapes. Due to this closed-form solution for eliminating translation, scale and rotation from the shape energy, no additional parameters enter the minimization. This prevents additional local minima and facilitates the minimization. On several segmentation tasks we were able confirm these effects by comparing the two approaches of optimizing explicit pose parameters and proposed implicit optimization. For example, for the problem presented in Figure 3.15, we did not manage to balance the minimization in a way that it converged to the desired segmentation by optimizing *explicit* pose parameters.

Contour evolution from initial to final with similarity invariant shape prior.



Further evolution upon switching off the statistical prior.

Figure 3.16: Segmentation of a partially occluded hand with and without shape prior in the simplified diffusion snake. The similarity invariant statistical shape prior permits a reconstruction of the hand silhouette in places where it is occluded (top row). Upon switching off the prior, the shape of the segmenting contour is no longer constrained and the contour evolution will simply maximize the homogeneity of the grey value in the separated regions (bottom row).

3.6.6 Dealing with Occlusion

The main idea of introducing the shape prior is that it is able to compensate for missing or misleading information. In the case of occlusion, for example, we expect the statistical shape prior to induce a *reconstruction of the shape silhouette* in parts of the image where the object is not visible.

This is demonstrated by the images in Figure 3.16, which show a hand covered by an artificially added occlusion. The top row shows the contour evolution from the initial to the final contour for the **SDS** with a similarity invariant shape prior. Note that the final contour correctly segments the hand in spite of the large occlusion. To further demonstrate the influence of the shape prior on the final segmentation, we simply switched off the prior by setting the weighting parameter $\alpha = 0$ in the total energy (3.31).

The bottom row in Figure 3.16 indicates the contour evolution without the statistical prior: The evolving contour simply separates light and dark regions, such that the object of interest is "lost", although the contour was optimally initialized. Of course, even with a statistical prior, the quality of the final segmentation slowly degrades as the size of the occlusion is increased. This is shown by a comparison with a larger and darker occlusion in Figure 3.17.

Initial No prior With prior No prior With prior

Figure 3.17: Increased occlusion. For two occlusions of different size and grey value, we compared segmentation results starting with the same initialization (left image), without and with statistical prior. We performed a gradient descent on the total energy (3.31) for the **SDS** (2.24) and the similarity invariant shape energy (3.23). The shape prior drastically improves segmentation results. However, depending on the initialization, the size and grey value of the occlusion, there may not be sufficient information left to correctly guide the contour evolution.

3.6.7 Dealing with Noise

A different case of missing information is given when the image containing the object of interest is corrupted by noise. Depending on the amount of noise, there may be very little information to drive the evolving contour towards the desired segmentation. Again, the statistical shape prior can improve segmentation, because it effectively reduces the dimension of the search space in such a way that segmentations which do not correspond to familiar shapes are ruled out a priori.

Figure 3.18, top left, shows the same input image as in Figure 3.12. However, this time, 75% of the pixels were replaced by an arbitrary grey value sampled from a uniform distribution over the interval $[0, 255]$. This means that only one of four pixels contains information about the input image. Figure 3.18 shows four steps in the contour evolution for the **SDS** with a similarity invariant shape prior. For the given initialization the segmentation process *with* prior is able to converge to the desired segmentation. In contrast, for the same initialization, the segmentation process *without* the shape prior fails to segment the object of interest, as shown in Figure 3.18, bottom right.

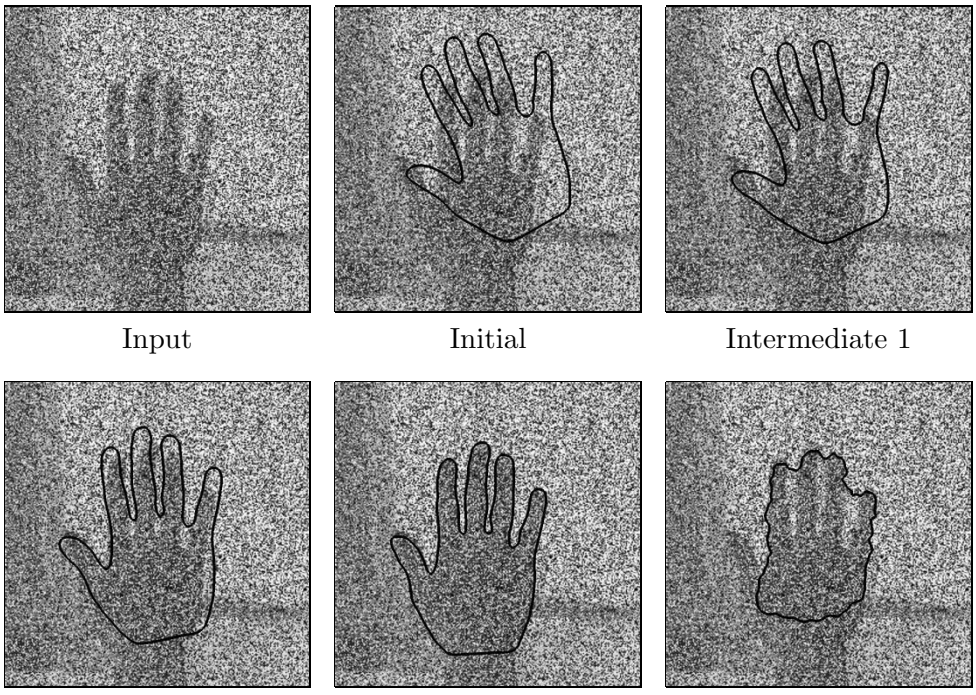


Figure 3.18: Segmentation of an image corrupted by noise. The input image is the one shown in Figure 3.12, with 75% of the pixels replaced by grey values sampled from a uniform distribution on the interval $[0, 255]$. Four frames from the gradient descent minimization indicate the contour evolution for the **SDS** with a similarity invariant shape prior. The frame on the bottom right shows the final segmentation obtained for the same initialization without a shape prior. By effectively suppressing unfamiliar shape deformations, the statistical prior facilitates convergence to the desired segmentation.

Chapter 4

Nonlinear Shape Statistics in Segmentation

4.1 Limitations of the Linear Model

In Section 3.3 we presented a model of shape statistics which is based on the assumption that the training shapes $\{z_i \in \mathbb{R}^{2N}\}$ are distributed according to a Gaussian probability density. Although this assumption tends to be fairly good in practice, it will fail as soon as the training set presents more complicated shape variation, which cannot be captured by second order statistics. This case occurs if several objects are included in the training set, or if one wants to "learn" the various silhouettes associated with different views of a 3D object. In this case one expects to find several clusters in shape space corresponding to the stable views of an object. Moreover, each cluster may by itself be quite non-Gaussian.

A standard way to verify the Gaussian hypothesis is to perform statistical tests such as the χ^2-test. In the following, we want to demonstrate the "non-Gaussianity" of a set of sample shapes in a different way, which gives a better intuitive understanding of the limitations of the Gaussian hypothesis in the context of shape statistics.

As an example, we again use a training set $\{z_i \in \mathbb{R}^{2N}\}$ of hand shapes, but this time it contains nine views of a right hand and nine views of a left hand, aligned as detailed in Section 3.1.4. Figure 4.1, left side, shows the training shapes projected onto the first two principal components and the level lines of constant energy[1] for the Gaussian model (3.17). Note that if the training set were Gaussian distributed, then all projections should be Gaussian distributed as well. Yet in the projection in Figure 4.1, left side, one can clearly distinguish two separate clusters containing the right hands (+) and the left hands (•). The images on the right of Figure 4.1 show sampling along the first three principal

[1] In this chapter, high dimensional data sets and the estimated energies will generally be visualized by projections onto the linear principal components, because these capture (by definition) the largest variation. Yet, it should be noted that in our case this projection suppresses 198 out of 200 dimensions. The depicted level lines only give a rough visualization of how the distribution of sample shapes is approximated, because the energy is determined in the plane spanned by the two principal components.

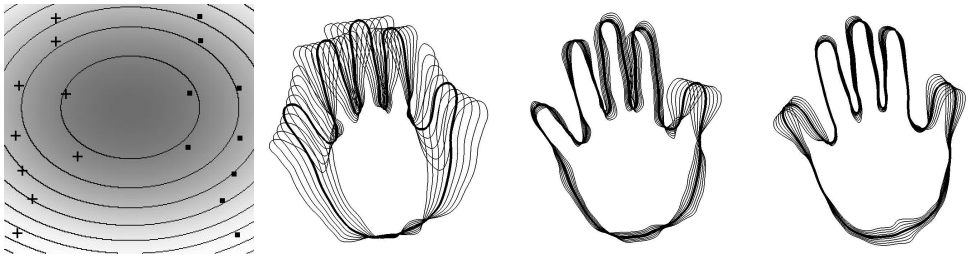


Figure 4.1: Left image: Training set containing 9 right hand (+) and 9 left hand (•) shapes in a projection onto the first two principal components. The estimated Gaussian model is visualized by shading and level lines of constant energy (3.17). **Right images:** Sampling up to two standard deviations along the first three principal components from the mean.

components from the mean, similar to what is shown in Figure 3.4 for the single-sided hand set.

As suggested by the level lines of constant energy, the first principal component — i.e. the mayor axis of the ellipsoid — corresponds to the deformation between right and left hands. This *morphing* from a left hand to a right hand is visualized in more detail in Figure 4.2. It shows that the Gaussian model tends to mix shapes belonging to different classes. Obviously the Gaussian model does not accurately represent the distribution of training shapes. In fact, according to the Gaussian model, the most probable shape is given by the mean shape, which is shown in the central image in Figure 4.2. Yet everyone would agree that this shape is not a valid hand. So in general, sampling along the different eigenmodes around the mean shape can give an intuitive feeling for the quality of the Gaussian assumption.

Figure 4.2: Sampling along the first principal component for a set containing right and left hands. Shapes of different classes are mixed in the Gaussian model. Note that according to the Gaussian model the mean shape (central image) is the most probable shape.

Once the training shapes are no longer Gaussian distributed, as in the above example, one needs to go beyond the linear models. We have chosen to introduce such nonlinearities in terms of *Mercer kernels*. Previous work in this field will be reviewed in the next two sections.

4.2 Mercer Kernel Methods

Based on the Mercer theorem [130, 49, 69], it is shown in [23] that for any continuous symmetric kernel $k(.,.)$ of a positive integral operator, one can construct a mapping ϕ into a space Y where k acts as a scalar product, i.e.:

$$k(x, y) = (\phi(x), \phi(y)) \,. \tag{4.1}$$

Conversely one can easily show that, given a continuous mapping ϕ, the equation (4.1) defines a continuous symmetric kernel of a positive integral operator. Such kernels are called *Mercer kernels*. In general, a given Mercer kernel k corresponds to an entire family of mappings ϕ.

This property of Mercer kernels can be exploited to model nonlinear transformations to a *feature space* in any algorithm for which the nonlinearity ϕ only appears in terms of scalar products. It is possible to model a whole family of nonlinearities by choosing a specific kernel. Moreover, the Mercer kernel approach permits to elegantly model nonlinear mappings into feature spaces Y of large (even infinite) dimension because the mapping ϕ is never evaluated explicitly.

The Mercer kernel approach has been extensively studied in such fields as feature extraction, classification or regression estimation (cf. [1, 180, 161, 27]). In this work, however, we intend to make use of it to construct a shape dissimilarity measure. For this purpose, we will employ the Mercer kernel approach to estimate the distribution of a set of sample points upon a nonlinear mapping ϕ to a feature space Y.

Our approach constitutes an extension of kernel PCA [164] to a probabilistic framework and was first proposed in [50]. More recently, it has also been suggested in [175].

4.3 Kernel Principal Component Analysis

In this section, we will introduce some notations[2] and briefly review results of kernel PCA [164], which is a particular Mercer kernel method.

4.3.1 Notation

Let $\chi = \{z_i\}_{i=1,\ldots,m}$ be a set of sample vectors $z_i \in \mathbb{R}^n$. Let

$$\phi : \mathbb{R}^n \to Y$$

be a (possibly nonlinear) map into a generally higher-dimensional feature space Y. Denote the mean of the mapped sample vectors by

$$\phi_0 := \frac{1}{m} \sum_{i=1}^{m} \phi(z_i), \tag{4.2}$$

[2]The following notations and the derivation of kernel PCA are based on [164]. We deviate only in so far as to incorporate the centering of the mapped training data into the derivation.

and the sample covariance matrix in Y by

$$\tilde{\Sigma} := \frac{1}{m} \sum_{i=1}^{m} \tilde{\phi}(z_i) \tilde{\phi}(z_i)^t, \tag{4.3}$$

where the notation

$$\tilde{\phi}(z) := \phi(z) - \phi_0 \tag{4.4}$$

was introduced to account for centering with respect to the mapped sample vectors in Y. We define the centered kernel by

$$\tilde{k} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \qquad \tilde{k}(x, y) := (\tilde{\phi}(x), \tilde{\phi}(y)). \tag{4.5}$$

Inserting definitions (4.4) and (4.2), it can be expressed in terms of the original kernel function (4.1):

$$\tilde{k}(x, y) = k(x, y) - \frac{1}{m} \sum_{k=1}^{m} (k(x, z_k) + k(y, z_k)) + \frac{1}{m^2} \sum_{k,l=1}^{m} k(z_k, z_l). \tag{4.6}$$

Moreover, we define the $m \times m$ kernel matrix K by:

$$K_{ij} := k(z_i, z_j) = (\phi(z_i), \phi(z_j)), \quad i, j = 1, \ldots, m, \tag{4.7}$$

and the *centered* kernel matrix $\tilde{K}$ by

$$\tilde{K}_{ij} := \tilde{k}(z_i, z_j) = (\tilde{\phi}(z_i), \tilde{\phi}(z_j)), \quad i, j = 1, \ldots, m. \tag{4.8}$$

From equation (4.6) it follows that

$$\tilde{K} = K - KE - EK + EKE, \qquad \text{where } E_{ij} = \frac{1}{m} \ \forall\, i, j = 1, \ldots, m. \tag{4.9}$$

4.3.2 PCA in Feature Space

The eigenvalues $\lambda_k > 0$ and eigenvectors $v_k \in Y \setminus \{0\}$ of the sample covariance matrix $\tilde{\Sigma}$ can be expressed in terms of the mapped sample vectors as follows.

Due to the definition of the covariance matrix, its eigenvectors v_k are part of the subspace $F \subset Y$ spanned by the centered mapped sample vectors:

$$v_k = \sum_{i=1}^{m} \alpha_i^k \, \tilde{\phi}(z_i). \tag{4.10}$$

It will now be shown, that the expansion coefficients α_i^k are related to the eigenvectors of the centered kernel matrix $\tilde{K}$. Since the eigenvectors v_k are part of the subspace F, the eigenvalue equation $\tilde{\Sigma} v_k = \lambda_k v_k$ is equivalent to its respective projections onto the centered sample points:

$$\tilde{\phi}(z_j)^t \, \tilde{\Sigma} \, v_k = \lambda_k \, \tilde{\phi}(z_j)^t \, v_k, \quad \forall j = 1, \ldots, m.$$

Using the centered kernel matrix $\tilde{K}$, one obtains:

$$\sum_{i,\ell=1}^{m} \tilde{K}_{j\ell} \tilde{K}_{\ell i} \, \alpha_i^k = m \lambda_k \sum_{i=1}^{m} \tilde{K}_{ji} \, \alpha_i^k, \quad \forall j = 1, \ldots, m,$$

or in matrix notation with $\alpha^k = (\alpha_1^k, \ldots, \alpha_m^k)^t$:

$$\tilde{K}^2 \alpha^k = m\lambda_k \tilde{K} \alpha^k.$$

The solutions of this equation are given by the solutions of the eigenvalue equation

$$\tilde{K}\alpha^k = m\lambda_k \alpha^k.$$

Let $\tilde{\lambda}_k$ be the eigenvalues of $\tilde{K}$ and $\tilde{\alpha}^k$ the corresponding normalized eigenvectors. Then the eigenvectors of the sample covariance matrix $\tilde{\Sigma}$ are given by $\lambda_k = m^{-1}\tilde{\lambda}_k$, and the eigenvector v_k is given by (4.10) with a coefficient vector $\alpha^k = \tilde{\lambda}_k^{-1/2}\tilde{\alpha}^k$. The latter normalization enforces the eigenvectors v_k to have unit length.

For a given input vector z, one obtains the nonlinear principal components (associated with the sample vectors $z_i \in \chi$ under the mapping ϕ) by projection onto the eigenvectors V^k in (4.10):

$$\big(V^k, \tilde{\phi}(z)\big) = \sum_{i=1}^{m} \alpha_i^k \big(\tilde{\phi}(z_i), \tilde{\phi}(z)\big) = \sum_{i=1}^{m} \alpha_i^k \, \tilde{k}(z_i, z). \tag{4.11}$$

With the relation (4.6), these nonlinear principal components can now be determined in terms of the Mercer kernel k corresponding to the mapping ϕ.

4.3.3 Feature Space Eigenmodes for Different Kernels

Rather than choosing an appropriate nonlinear mapping ϕ, one chooses an appropriate kernel function k which corresponds to an entire family of possible nonlinearities via the identity (4.1).

Common choices of the kernel function are the *Gaussian kernel*

$$k(x, y) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{|x-y|^2}{2\sigma^2}\right), \quad x, y \in \mathbb{R}^n, \tag{4.12}$$

where the normalizing factor was introduced for future purposes, the *homogeneous* and *inhomogeneous polynomial kernels*

$$k(x, y) = (x^t y)^d, \qquad k(x, y) = (x^t y + 1)^d, \tag{4.13}$$

where the degree d is a positive integer, and the *sigmoid kernel*

$$k(x, y) = \tanh\big(a(x^t y) + b\big), \tag{4.14}$$

with parameters a and b.

In the following, we will visualize the first few kernel principal components of a data set in $\mathbb{R}^2$, for the Gaussian kernel with two values of σ and the homogeneous polynomial kernel with three different degrees d.

Figure 4.3 shows the data set consisting of random samples from three clusters in the domain $[-1, 1]^2$, and theprojections onto the first few kernel principal components for three homogeneous polynomial kernels with degrees $d = 1$, $d = 2$ and $d = 4$, respectively. The case $d = 1$ is equivalent to a linear

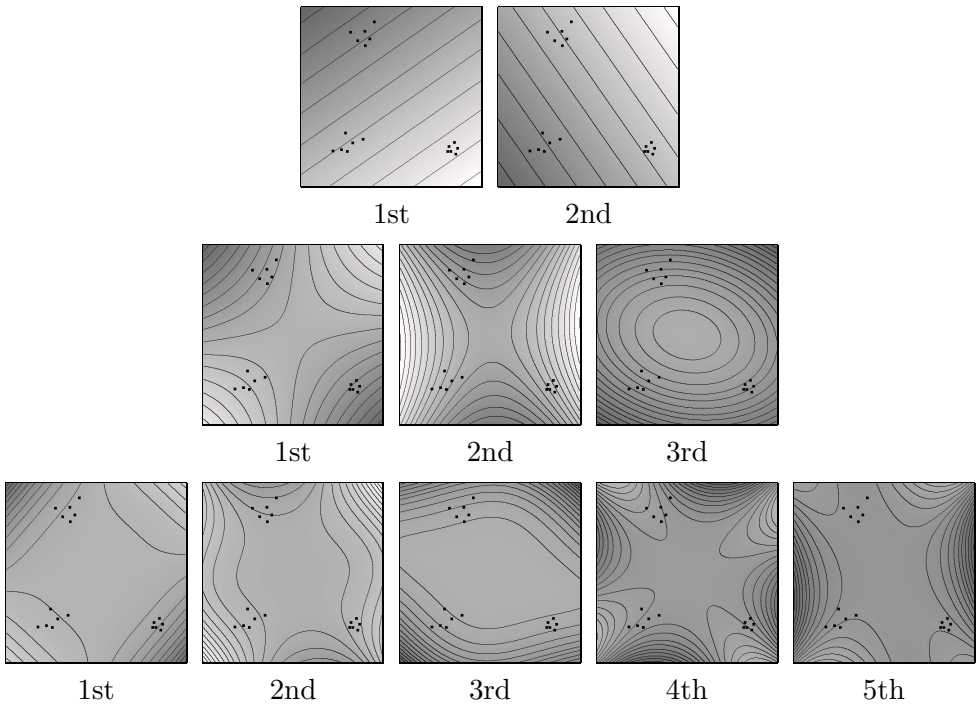


Figure 4.3: Projections onto the first kernel principal components for the homogeneous polynomial kernel (4.13). The degrees are $\boldsymbol{d = 1}$ (**top**), $\boldsymbol{d = 2}$ (**middle**) and $\boldsymbol{d = 4}$ (**bottom**). Note that the case $d = 1$ is identical with the linear principal component analysis.

principal component analysis, which means that linear PCA can be considered as a specific case of kernel PCA. For a given test point, these projections can be used to determine its class membership.

Figure 4.4, top, shows projections onto the first 6 kernel principal components for the Gaussian kernel (4.12) with a width of $\sigma = 0.5$. The various projections each separate different clusters or subdivide the individual clusters. Therefore a given sample point can be classified according to its projections onto the various kernel principal components.

The width σ of the Gaussian kernel (4.12) represents the spatial scale at which clusters are separated. Similar projections onto the kernel principal components for a Gaussian kernel of smaller width $\sigma = 0.1$ are shown in Figure 4.4, bottom. Again the clusters are separated by projections onto the feature space eigenvectors. Yet the absolute value of the projection decreases more rapidly with the distance from the data clusters.

In contrast to the projections for the polynomial kernel, the clusters for the Gaussian kernel correspond to extremal values of the respective projections. This indicates that the Gaussian kernel is more adapted to the problem of clustering and modeling data dissimilarity measures. A more mathematical explanation for the advantages of such stationary (or translation-invariant) kernel functions will be given in Appendix C, where we relate the Mercer kernel approach to classical methods of density estimation.

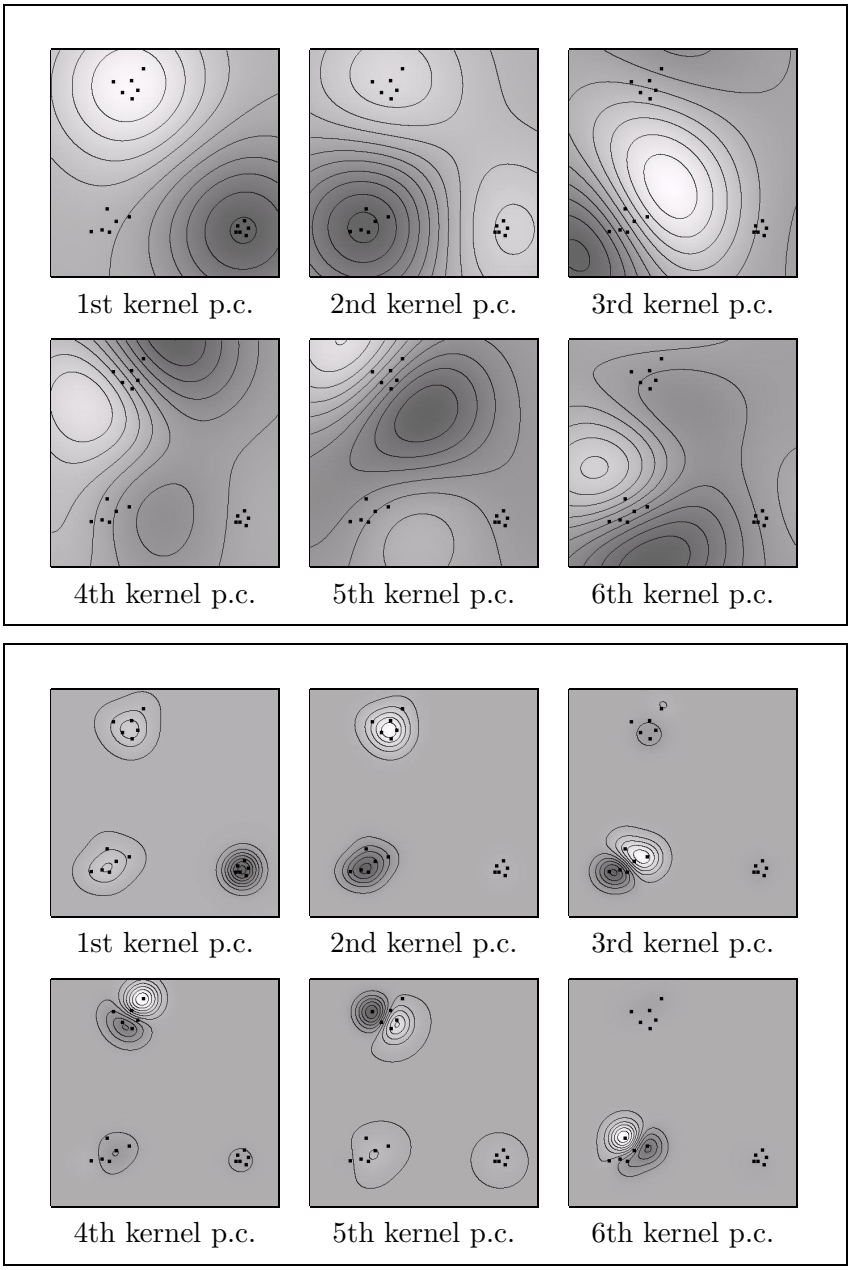


Figure 4.4: Projection onto the first **6** kernel principal components for the Gaussian kernel (4.12) with $\boldsymbol{\sigma = 0.5}$ **(top)** and $\boldsymbol{\sigma = 0.1}$ **(bottom)**. The value of the projection (4.11) onto the respective kernel principal component is visualized by level lines and shading, where dark and light areas correspond to the extrema of each projection. Note that the various kernel principal components separate the clusters and subclusters. A given point in input space can now be classified according to these projections: The three clusters can be separated by the first two projections, whereas the other projections permit a further subdivision of the clusters.

At this point, however, we will not go into detail about the meaning of the different choices of the kernel function. In fact, we will only consider the Gaussian kernel (4.12) in the following.

As we have seen, kernel PCA can be employed for classification and feature extraction (cf. [163, 194]). In the following, we will make use of it to construct a dissimilarity measure between shape vectors after mapping them to a feature space Y with a nonlinear function ϕ.

4.4 Probabilistic Modeling in Feature Space

In the following, we will model the distribution $\mathcal{P}$ of the mapped sample vectors in the feature space Y. As in all Mercer kernel methods, the mapping ϕ will be modeled implicitly by a kernel function k, for which we will use the Gaussian kernel (4.12). In particular, we will derive the energy $E = -\log \mathcal{P} + \text{const}$. We will discuss the relation to kernel PCA and present a heuristic estimate for the kernel width σ. A more detailed study of the proposed energy and its relation to classical methods of density estimation is postponed to Appendix C, so as not to break the flow of the argument.

4.4.1 The Feature Space Gaussian

Just as in the linear case — see the discussions in Section 3.3 — the kernel PCA approach can be extended to a probabilistic framework.

Let $\chi = \{z_i \in \mathbb{R}^n\}$ be a set of training shapes, where $n = 2N$, N being the number of spline control points. Assume they are aligned with respect to similarity transformations and cyclic permutation of the control points, as discussed in Section 3.1.4. In analogy to the Gaussian model in $\mathbb{R}^n$ presented in Section 3.3, we now assume that the *mapped* training shapes are distributed according to a Gaussian density in the feature space Y. Figure 4.5 shows a schematic drawing of the original space and the mapping to the feature space Y. The linear subspace spanned by the mapped training vectors is denoted by F, its orthogonal complement in Y is denoted by $\overline{F}$.

The energy associated with this Gaussian probability density in feature space is given by:

$$\boxed{E_\phi(z) \;=\; \tilde{\phi}(z)^t \;\; \Sigma_\phi^{-1} \;\tilde{\phi}(z)\,,} \tag{4.15}$$

where $\tilde{\phi}(z) = \phi - \phi_0$, where ϕ_0 is given by the mean of the mapped vectors. As in the linear case, this shape energy is a quadratic function — see equation (3.17). This time, however, it is not quadratic in the input vector z, but quadratic in the mapped vector $\phi(z)$. As we will see later on, the respective estimates in the original space are fundamentally different.

As discussed in the linear case — see Section 3.3 — the estimated covariance matrix $\hat{\Sigma}$ is generally not invertible, since the mapped sample vectors only span a linear subspace F in Y. We therefore revert to a regularized covariance matrix Σ_ϕ, where the zero eigenvalues are replaced by a constant $\lambda_\perp$:

$$\Sigma_\phi = V\,\Lambda\,V^t \;+\; \lambda_\perp \left(I \;-\; V\,V^t\right). \tag{4.16}$$

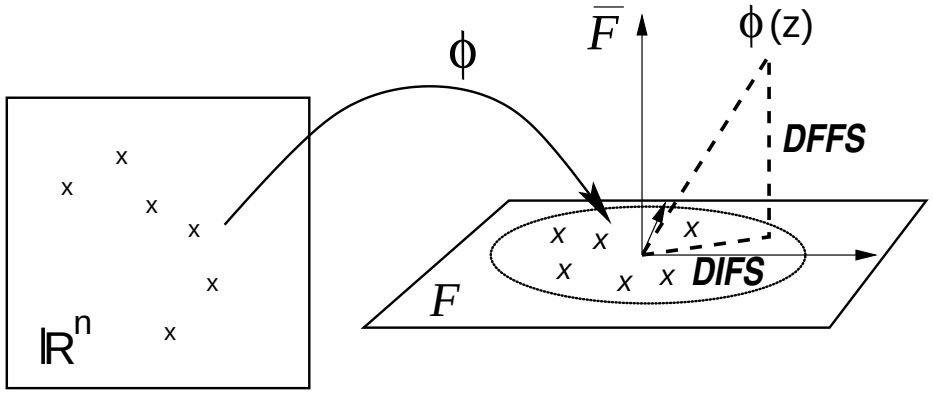


Figure 4.5: Schematic diagram of the nonlinear mapping ϕ of the training shapes from the original shape space $\mathbb{R}^n$ to a generally higher-dimensional space $Y = F \bigoplus \overline{F}$. The probabilistic distance of a mapped point $\phi(z)$ to the mapped training points can be decomposed into a distance from the feature space F (DFFS) and a distance in the feature space (DIFS).

The diagonal matrix Λ contains the nonzero eigenvalues $\lambda_1 \geq \cdots \geq \lambda_r$ of $\hat{\Sigma}$, and V is the matrix of the corresponding eigenvectors $v_1, \ldots, v_r$.

Inserting (4.16) splits energy (4.15) into two terms:

$$E_\phi(z) \;=\; \sum_{k=1}^{r} \lambda_k^{-1} \left(v_k, \tilde{\phi}(z)\right)^2 \;+\; \lambda_\perp^{-1} \left(|\tilde{\phi}(z)|^2 \;-\; \sum_{k=1}^{r} \left(v_k, \tilde{\phi}(z)\right)^2 \right) . \qquad (4.17)$$

With expansion (4.10), we obtain the final expression for our energy:

$$E_\phi(z) = \sum_{k=1}^{r} \left(\sum_{i=1}^{m} \alpha_i^k \, \tilde{k}(z_i, z) \right)^2 \cdot \left(\lambda_k^{-1} - \lambda_\perp^{-1}\right) \;+\; \lambda_\perp^{-1} \cdot \tilde{k}(z, z) \, . \qquad (4.18)$$

As in the case of kernel PCA, the nonlinearity ϕ only appears in terms of the kernel function. This allows to specify an entire family of possible nonlinearities by the choice of the associated kernel function. In the following, we will use the Gaussian kernel (4.12) for k, the centered kernel $\tilde{k}$ being given by equation (4.6). For a justification of this choice, we refer to Appendix C.

4.4.2 Relation to Kernel PCA

Just as in the linear case — see Section 3.3 and [131] — the regularization (4.16) of the covariance matrix causes a splitting of the energy into two terms (4.17), which can be considered as a *distance in feature space*[3]:

$$\text{DIFS} = \sum_{k=1}^{r} \lambda_k^{-1} \left(v_k, \tilde{\phi}(z)\right)^2 , \qquad (4.19)$$

[3]We want to point out, that in order to adhere to the corresponding literature, the term "feature space" is used inconsistently: Sometimes it refers to the full space Y, whereas sometimes it refers to the subspace $F \subset Y$, which is spanned by the mapped training vectors.

and a *distance from feature space*:

$$\text{DFFS} = \lambda_\perp^{-1} \left(|\tilde{\phi}(z)|^2 - \sum_{k=1}^{r} \left(v_k, \tilde{\phi}(z) \right)^2 \right) . \tag{4.20}$$

Both of these distances[4] are visualized in Figure 4.5.

For the purpose of pattern reconstruction in the framework of kernel PCA, it was suggested to minimize a reconstruction error [162], which is identical with the DFFS. This approach is based on the assumption that the entire plane spanned by the mapped training data corresponds to acceptable patterns. However, we believe that this is not a valid assumption: Already in the linear case, moving too far along an eigenmode will produce patterns which have almost no similarity to the training data, although they are still accepted by the hypothesis. Moreover, the distance DFFS is not based on a probabilistic model. In contrast to that, energy (4.18) is derived from a Gaussian probability distribution. It minimizes both the DFFS and the DIFS.

4.4.3 On the Regularization of the Covariance Matrix

A regularization of the covariance matrix in the case of kernel PCA — as done in (4.16) — was first proposed in [50] and has also been suggested more recently in [175] under the name of *probabilistic feature–space PCA*.

The choice of the parameter $\lambda_\perp$ is not a trivial issue. As discussed in Section 3.3.2, such regularizations of the covariance matrix have been proposed for the linear case. There [131, 176], the constant $\lambda_\perp$ is estimated as the mean of the replaced eigenvalues by minimizing the Kullback-Leibler distance of the two densities corresponding to the sample covariance matrix and its regularized version. However, we believe that this is not neccessarily the optimal choice of the regularization constant $\lambda_\perp$. The Kullback-Leibler distance is supposed to measure the error with respect to the correct density, which means that the sample covariance matrix calculated from the training data is assumed to be the correct one. Yet this is not the case because the number of training points is limited. For essentially the same reason this approach does not extend to the nonlinear case considered here[5]: Depending on the type of nonlinearity ϕ, the covariance matrix is potentially infinite-dimensional such that the mean over all replaced eigenvalues will be zero for any finite number of training samples.

As in the linear case, we therefore propose to choose

$$0 < \lambda_\perp < \lambda_r,$$

which means that unfamiliar variations from the mean (in feature space) are less probable than the smallest variation observed on the training set. We fix $\lambda_\perp = \lambda_r/2$ in all practical applications.

[4] In precise terminology DIFS and DFFS are *squared* distances.

[5] In [175], $\lambda_\perp = 0.25^2$ is fixed arbitrarily, which deviates from the choice (3.18), which the same authors proposed in the linear case.

4.4.4 On the Choice of the Hyperparameter σ

The last parameter to be fixed in the proposed energy (4.18) is the hyperparameter σ in the definition of the Gaussian kernel (4.12). Let μ be the average distance between two neighboring data points:

$$\mu^2 := \frac{1}{m} \sum_{i=1}^{m} \min_{j \neq i} |z_i - z_j|^2 . \tag{4.21}$$

In order to get a smooth energy landscape, we propose to choose σ in the order of μ. In practice we used

$$\sigma = 1.5\,\mu$$

for most of our experiments. We chose this somewhat heuristic measure for the following favorable properties: μ is insensitive to the distance of clusters as long as each cluster contains more than one data point, it scales linearly with the data points, and it is robust with respect to perturbation of the data points.

If outliers the training set contains outliers, i.e. clusters with only one sample, one could refer to the more robust L_1-norm or more elaborate robust estimators in (4.21). Alternatively it could be estimated by cross validation. Since the optimal estimation of the kernel width σ is not the focus of our contribution, we will not elaborate on these issues. A further justification for the above choice of the kernel width will be given at the end of Appendix C.2.

4.5 Density Estimate for Silhouettes of 3D Objects

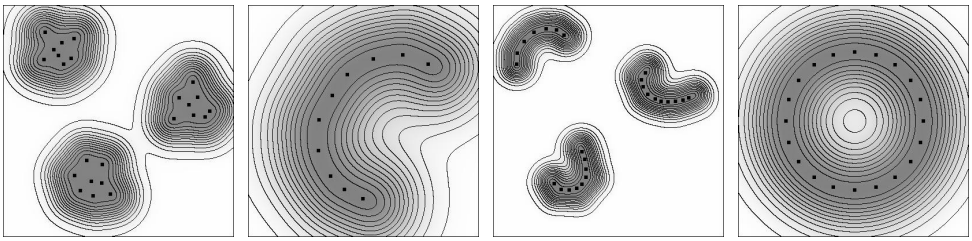


Figure 4.6: Density estimate (4.15) for artificial 2D data. Distributions of variable shape are well estimated by the Gaussian hypothesis in feature space. We used the kernel (4.12) with $\sigma = 1.5\,\mu$.

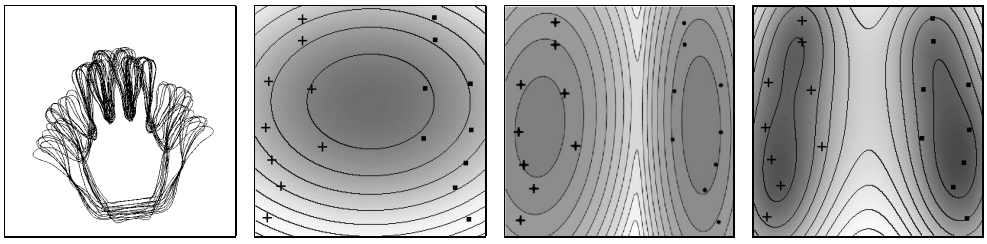
Although energy (4.15) is quadratic in the space Y of mapped points, it is generally not convex in the original space, showing several minima and level lines of essentially arbitrary shape. Figure 4.6 shows artificial 2D data and the corresponding lines of constant energy $E_\phi(z)$ in the original space. These indicate that the Gaussian in feature space fundamentally differs from a Gaussian in the original space: The proposed energy can capture distributions corresponding to several clusters of data points — which may be the case if several objects are included in the training set. Moreover, each individual cluster does not need to be ellipsoidal, as must be the case for the model of mixtures of

Gaussians. For an explanation of this behavior, we refer to Appendix C.

Since the input dimension of the data does not play an important role in the derivation of the energy (4.18), we expect to find a similar capacity to model arbitrary point distributions in higher dimensions as well.

In our application, we automatically fit a closed quadratic spline curve around each object in a set of binarized views. All spline curves have N=100 control points, set equidistantly. The corresponding polygons defined by the control points $z = (x_1, y_1, \ldots, x_N, y_N)$ are aligned with respect to translation, rotation, scaling and cyclic permutation — see Section 3.1.4. The resulting data was then used to determine the density estimate $E_\phi(z)$ in (4.18).

For the visualization of the density estimate and the training shapes, all data was projected onto two of the principal components e_a and e_b of a linear PCA. Note that due to the projection, this visualization only gives a very rough sketch of the true distribution in the 200-dimensional shape space. The energy level lines are determined for the plane spanned by these two eigenvectors, i.e. for the points $z = \lambda_a e_a + \lambda_b e_b$, with different values of λ_a and λ_b. Therefore, the correspondence of data points and energy level lines is only true up to variation in the remaining eigenvector components.



Aligned contours | Simple Gaussian | Mixture model | Feature space Gaussian

Figure 4.7: Model comparison. Density estimates for a set of left (•) and right (+) hands, projected onto the first two principal components. **From left to right:** Aligned contours, simple Gaussian, mixture of Gaussians, Gaussian in feature space (4.15). Both the mixture model and the Gaussian in feature space capture the two-class structure of the data. However, the estimate in feature space is unsupervised and produces level lines which are not necessarily elliptical.

Figure 4.7 shows density estimates for a set of right hands and left hands. The estimates correspond to the hypotheses of a simple Gaussian and a mixture of Gaussians in the original space, and a Gaussian in feature space. Although both the mixture model and our estimate in feature space capture the two distinct clusters, there are several differences: Firstly, the mixture model is supervised — the number of classes and the class membership must be known — and secondly, it only allows level lines of elliptical shape, corresponding to the hypothesis that each cluster by itself is a Gaussian distribution. The model of a Gaussian density in feature space does not assume any prior knowledge

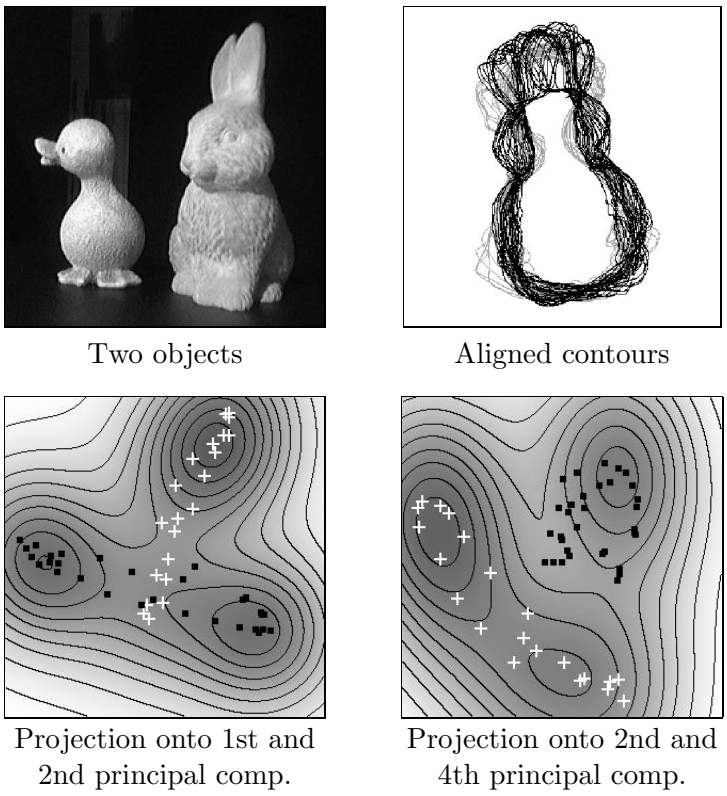


Figure 4.8: Density estimate for views of two 3D objects. The training shapes of the duck (white +) and the rabbit (black •) form distinct clusters in shape space which are well captured by the energy level lines shown in appropriate 2D projections.

and produces level lines which capture the true distribution of the data even in the case that it does not correspond to a sum of hyperellipsoids.

This is demonstrated on a set of training shapes which correspond to different views of two 3D objects. Figure 4.8 shows the two objects, their contours after alignment and the level lines corresponding to the estimated energy density (4.15) in appropriate 2D projections. In particular, the projection on the second and fourth principal component shows that the clusters associated with the two objects do not overlap.

To demonstrate how the width σ of the kernel function (4.12) influences the estimated density, Figure 4.9 shows the projected density plots of the energy (4.18) for several values of σ. The set of training shapes contains four right and four left hands. With decreasing granularity σ, the formation of more and more clusters is facilitated.

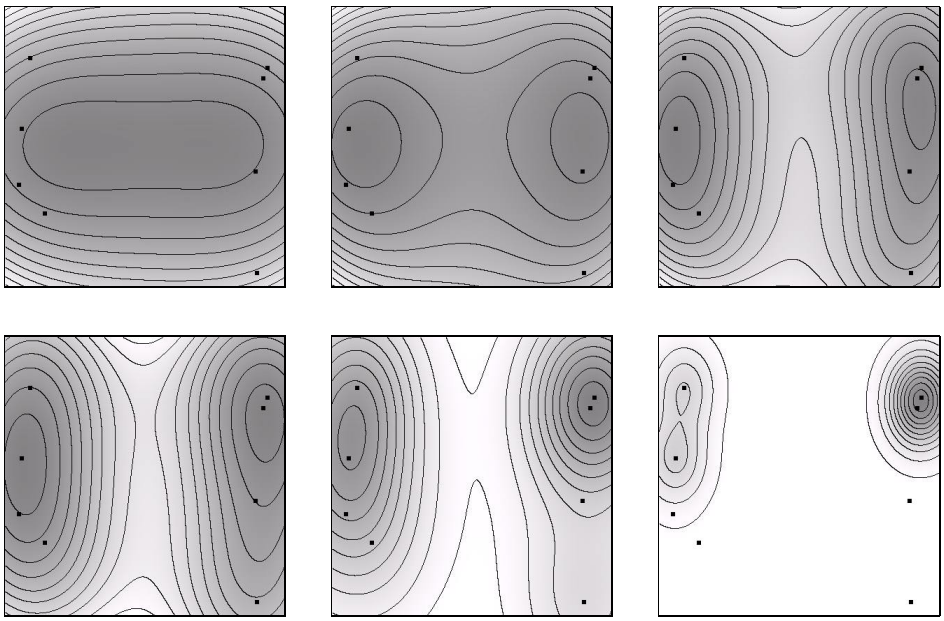


Figure 4.9: Influence of the kernel width on the energy. The figures show the training set of four right and four left hands and the estimated energy (4.18) for decreasing values of the kernel width σ. The kernel width determines the spatial scale for the cluster formation.

4.6 Nonlinear Shape Statistics in Segmentation

In the previous sections, we have constructed a shape prior based on a nonlinear mapping ϕ of the training shapes to a feature space Y and the hypothesis that the mapped training vectors are distributed according to a Gaussian density in this feature space Y.

In this section, we will combine this nonlinear shape prior with the diffusion snakes introduced in Sections 2.4 and 2.5. The resulting segmentation approach will drive the contour to segment a given input image by taking into account both the image information and the statistical shape knowledge defined via the Gaussian in feature space.

Let $\{z_i\}$ be a set of training shapes, aligned as discussed in Section 3.1.4. We propose to minimize the total energy given by:

$$E(z) = E_{image}\,(u, C_z) \;+\; \alpha\, E_\phi(\hat{z}), \tag{4.22}$$

where the image energy is (in our case) given by the simplified diffusion snake (2.24), and E_ϕ is the nonlinear shape energy (4.15) associated with the Gaussian model in feature space. The argument $\hat{z}$ denotes the control point vector z after optimal alignment with respect to the mean of the training data:

$$\hat{z} = \frac{R_\theta\, z_c}{|R_\theta\, z_c|},$$

where R_θ denotes the optimal rotation of the centered control point polygon z_c with respect to the mean shape $\bar{z}$. This prior alignment of the argument of E_ϕ provides invariance of the shape prior with respect to similarity transformations — see Section 3.4.2 for details.

As in Sections 2.5 and 3.5, we minimize the total energy (4.22) by gradient descent to obtain a segmentation of a given input image, which at the same time maximizes the grey value homogeneity in the regions separated by the contour, and minimizes the shape dissimilarity measure, defined in terms of the feature space energy E_ϕ and the intrinsic alignment with respect to similarity transformations.

The gradient descent equations for the control point (x_m, y_m) are given by:

$$\begin{aligned}
\frac{dx_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \left[\left(e^+(s_i,t) - e^-(s_i,t)\right) n_x(s_i,t) \;+\; \nu \left(x_{i-1} - 2x_i + x_{i+1}\right) \right] \\
&\quad - \alpha \left[\frac{dE_\phi(\hat{z})}{d\hat{z}} \, \frac{d\hat{z}}{dz} \right]_{2m-1}, \\
\frac{dy_m(t)}{dt} &= \sum_{i=1}^{N} \left(\mathbf{B}^{-1}\right)_{mi} \left[\left(e^+(s_i,t) - e^-(s_i,t)\right) n_y(s_i,t) \;+\; \nu \left(y_{i-1} - 2y_i + y_{i+1}\right) \right] \\
&\quad - \alpha \left[\frac{dE_\phi(\hat{z})}{d\hat{z}} \, \frac{d\hat{z}}{dz} \right]_{2m}.
\end{aligned} \tag{4.23}$$

They extend the contour evolution equations in (2.30) by the last term, which maximizes the similarity of the evolving contour with respect to the set of training shapes.

The three terms in each of the equations in (4.23) can be interpreted as follows (cf. Section 3.5):

- The first term forces the contour towards the object boundaries, by maximizing a homogeneity criterion in the adjoining regions, which compete in terms of their energy densities e^+ and e^-.

- The second term enforces an equidistant spacing of control points, thus minimizing the length measure (2.21). This prevents the formation of cusps during the contour evolution.

- The last term in (4.23) maximizes the similarity of the evolving contour with respect to the set of training shapes. It consists of two components: The first one,

  $$-\frac{dE_\phi(\hat{z})}{d\hat{z}}, \tag{4.24}$$

  is the negative gradient on the energy (4.18) evaluated at the aligned vector $\hat{z}$. It forces the aligned contour to descend into the nearest minimum of the feature space energy — as depicted in the energy plots of the previous section. The second component,

  $$\frac{d\hat{z}}{dz},$$

arises due to the similarity invariant formulation of the energy. It accounts for the rotation, translation and scaling of the contour, as explained in Section 3.4.2.

For completeness, we will present the formulas for the gradient (4.24) of the kernel energy (4.18). It is given by:

$$\begin{aligned}\frac{dE_\phi(z)}{dz} &= 2\sum_{k=1}^{r}\left(\sum_{i=1}^{m}\alpha_i^k\,\tilde{k}(z_i,z)\right)\left(\sum_{j=1}^{m}\alpha_j^k\,\frac{d\tilde{k}(z_j,z)}{dz}\right)\left(\lambda_k^{-1}-\lambda_\perp^{-1}\right)\\ &\quad+\lambda_\perp^{-1}\tilde{k}(z,z)\frac{d\tilde{k}(z,z)}{dz}\,.\end{aligned}$$

The two gradients of the centered kernel function (4.6) are given by:

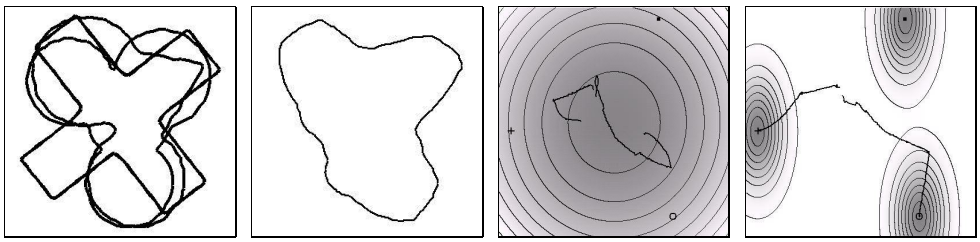
$$\begin{aligned}\frac{d\tilde{k}(z_j,z)}{dz} &= k(z,z_j)\,\frac{(z_j-z)}{\sigma^2}-\frac{1}{M}\sum_{l=1}^{M}k(z_l,z)\,\frac{(z_l-z)}{\sigma^2}\\ \frac{d\tilde{k}(z,z)}{dz} &= -\frac{2}{M}\sum_{l=1}^{M}k(z_l,z)\,\frac{(z_l-z)}{\sigma^2},\end{aligned}$$

for the case of the Gaussian kernel (4.12).

4.7 Numerical Results

In this section, we present a number of experimental results obtained by combining the diffusion snakes with the nonlinear shape prior, as detailed in the previous section.

In contrast to the linear shape prior, the nonlinear one permits the formation of several local minima corresponding to different objects or different stable views of a 3D object. Therefore, we generally iterate the gradient descent without the prior until stationarity of the contour first. Then we align the obtained contour to the mean of the training shapes with respect to similarity transformation and renumbering of the control points, before we include the prior and iterate until convergence. This guarantees that a maximum of information is extracted from the image before the prior "decides" which of the respective minima is the appropriate one. Obviously, the performance of this approach is rather sensitive to the initial alignment of the segmenting contour before the introduction of the prior. If the occlusion of a given object is too large, then the contour may be aligned incorrectly and the prior will fail to improve the segmentation. In this case, global optimization schemes might represent a remedy. However, they have not been evaluated in this work. On the other hand, local minimization schemes have several advantages: Firstly, they tend to be much faster — especially for optimization problems of high dimension. With 100 control points and variables u_i for the mean grey value in each region, we work in more than 200 dimensions. Secondly, the local method works well in applications such as tracking, where the segmentation of one image frame is generally a good initialization for segmenting the next frame. This will be demonstrated in Section 4.7.4.



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

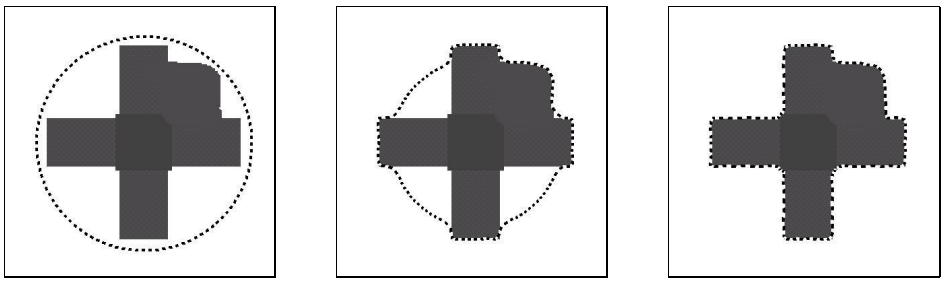
Figure 4.10: Linear versus nonlinear shape prior. **(a)** Aligned silhouettes of three training shapes. **(b)** Mean shape. **(c)** Density plot for the Gaussian model in shape space. **(d)** Density plot for the Gaussian model in feature space. Both density plots contain the path of the segmenting contour corresponding to the segmentation processes shown in Figures 4.11 and 4.12, respectively.

4.7.1 Linear versus Nonlinear Shape Prior

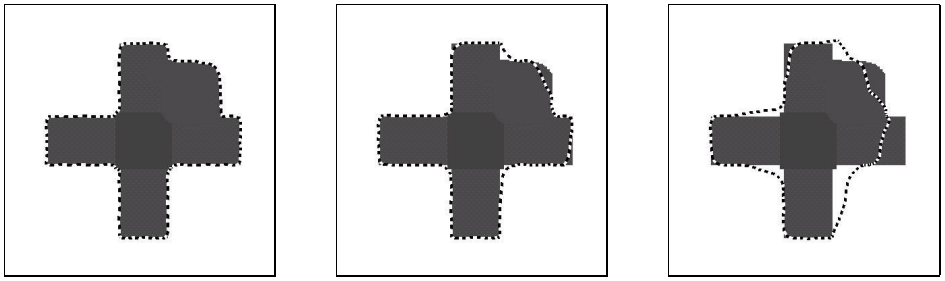
In the first example, we compare segmentation results obtained with the linear shape prior, as explained in Section 3.5, to those obtained with the nonlinear shape prior. For simplicity, we will only use three training shapes. The aligned training shapes and the respective mean contour are shown in Figure 4.10, (a) and (b). The mean shape corresponds to a morphing of the three shapes. Since these are fairly different, the mean shape does not resemble any of the individual shapes. This by itself is an indication that the Gaussian model in shape space is not an adequate probabilistic model for the given training set. The three training shapes and the estimated energies are shown in Figure 4.10 for the linear model (c), and for the nonlinear one (d).

Figure 4.11 shows segmentation results obtained for a partly occluded image of one of the objects. We first iterated the segmentation process without a prior until stationarity of the contour (top right image). Starting with this initialization we aligned the contour with respect to the training set and iterated the segmentation process, once with the linear prior (middle row) and once with the nonlinear prior (bottom row). The linear prior tends to pull the segmenting contour towards the mean of the training shapes, which is the most probable shape in the Gaussian model — see Figure 4.10, (c). In contrast, the nonlinear shape energy comprises three minima corresponding to the three different objects. Therefore, upon introduction of the nonlinear prior, the segmenting contour is drawn into the nearest minimum corresponding to the object of interest. In contrast to the linear prior, the nonlinear one permits a segmentation of the object of interest which ignores the prominent occlusion.

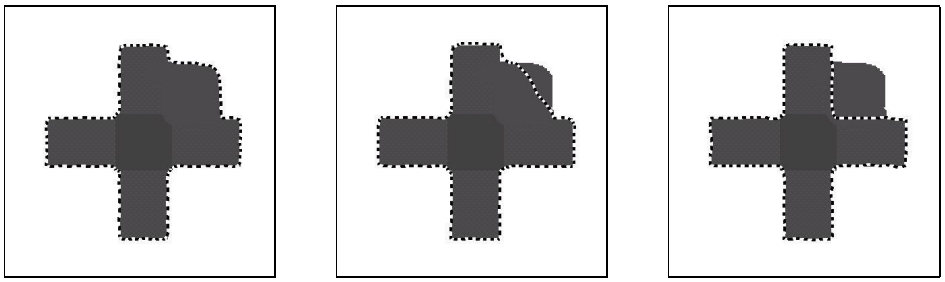
Figure 4.12 shows that a similar result is obtained with the *same* prior for an image of another object from Figure 4.10. In particular, the bottom row shows the similarity invariance of the prior: Upon introduction of the prior, the contour initially rotates, which seems to be energetically favorable in the beginning. Without presenting further evidence, we note that similar



Contour evolution **without prior**.



Evolution upon introduction of the **linear prior**.

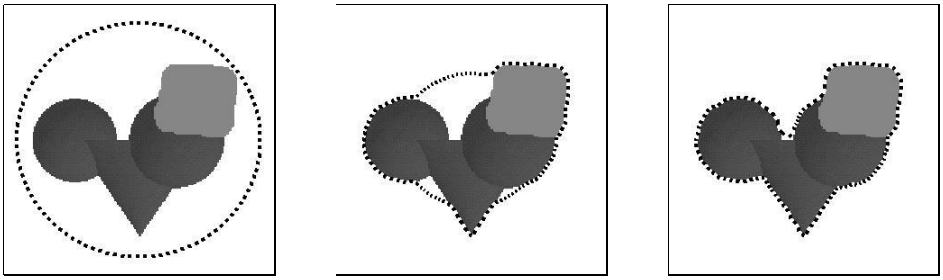


Evolution upon introduction of the **nonlinear prior**.

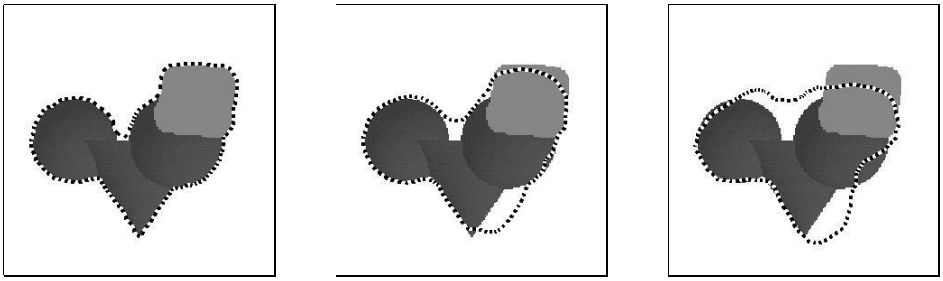
Figure 4.11: Segmentation of a partly occluded image of the first object from figure 4.10. While the linear prior tends to pull the contour towards the mean shape, the nonlinear one clearly associates the contour with one of the training objects. The paths of the two contour evolutions with the linear and the nonlinear prior are shown in a projection into the respective energy plots in Figure 4.10, (c) and (d).

segmentation properties can be demonstrated for partly occluded images of the third object in the training set.

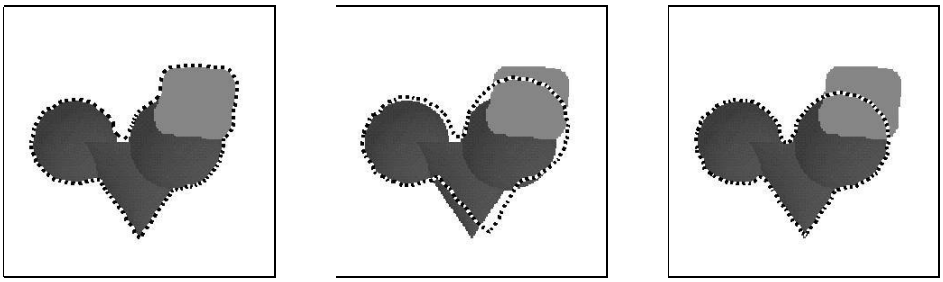
In order to visualize the effect of the prior on the segmentation process, we projected the path of the evolving contour into the energy density plots in Figure 4.10, (c) and (d). While the linear prior draws the contour towards the center of the distribution (i.e. towards the mean shape), the nonlinear prior drives the contour towards the nearest of *several* local minima. This property permits to encode shapes of different classes in a *single* nonlinear shape prior.



Contour evolution **without prior**.



Evolution upon introduction of the **linear prior**.



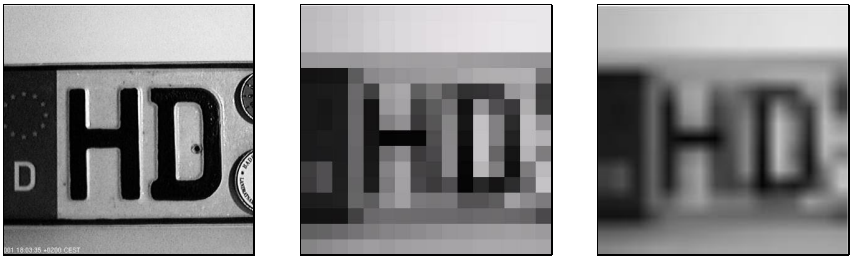
Evolution upon introduction of the **nonlinear prior**.

Figure 4.12: Segmentation of a partly occluded image of the second object from Figure 4.10. In contrast to the linear prior, the nonlinear one does not mix the 3 objects, such that a reconstruction of each of them is possible. The projected contour paths in Figure 4.10, (c) and (d), show that the linear prior draws the contour towards the center of the three objects, whereas the nonlinear one draws it to one the nearest one of the three learned shapes. Note that we used the same priors as in Figure 4.11.

4.7.2 Encoding Several Training Objects

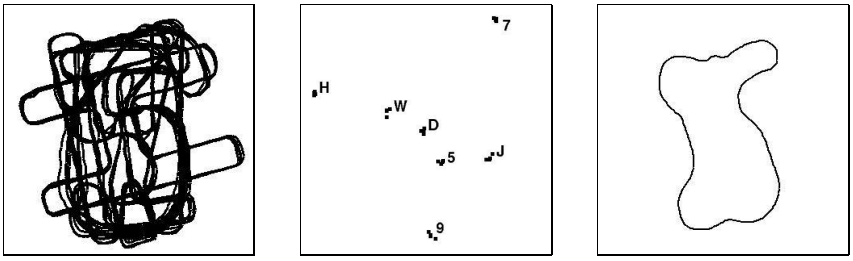
The following example is an application of our method which shows how the nonlinear shape prior can encode a number of different alphabetical letters and thus improve the segmentation of these letters in a given image.

We want to point out that there exists a vast number of different methods for optical character recognition. We do not claim that the present method is optimally suited for this task, and we do not claim that it outperforms existing methods. The following results only show that our rather general segmentation



(a) **(b)** **(c)**

Figure 4.13: **(a)** Original image region of 200×200 pixels. **(b)** Subsampled to 16×16 pixels. **(c)** Subsampled image upon bilinear smoothing.



(a) **(b)** **(c)**

Figure 4.14: **(a)** Aligned training shapes. **(b)** Projection onto the first and third (linear) principal component. **(c)** Mean shape.

approach with the nonlinear shape prior can be applied to a large variety of very different tasks.

A set of 7 letters and digits were segmented (several times) without any shape prior in an input image as the one shown in Figure 4.13, (a). The obtained contours were used as a training set to construct the shape prior. Figure 4.14 shows the set of aligned contours and their projection into the plane spanned by the first and third principal component (of a linear PCA). The clusters are labeled with the corresponding letters and digits. Again, the mean shape, shown in 4.13, (c), indicates that the linear model is not an adequate model for the distribution of the training shapes.

In order to generate a realistic task, we subsampled the input image to a resolution of 16×16 pixels, as shown in Figure 4.13, (b). Such a low resolution is a common problem in digital image processing. We then presmoothed the subsampled image by bilinear filtering as shown in Figure 4.13, (c).

Given such an input image, we initialized the contour, iterated the segmentation process without prior until stationarity and then introduced either the linear or the nonlinear shape prior. Figure 4.15 shows segmentation results without prior, with the linear prior and with the nonlinear prior. Again, the convergence of the segmenting contour towards one of the learnt letters is visualized by appropriate projections onto the first two linear principal components

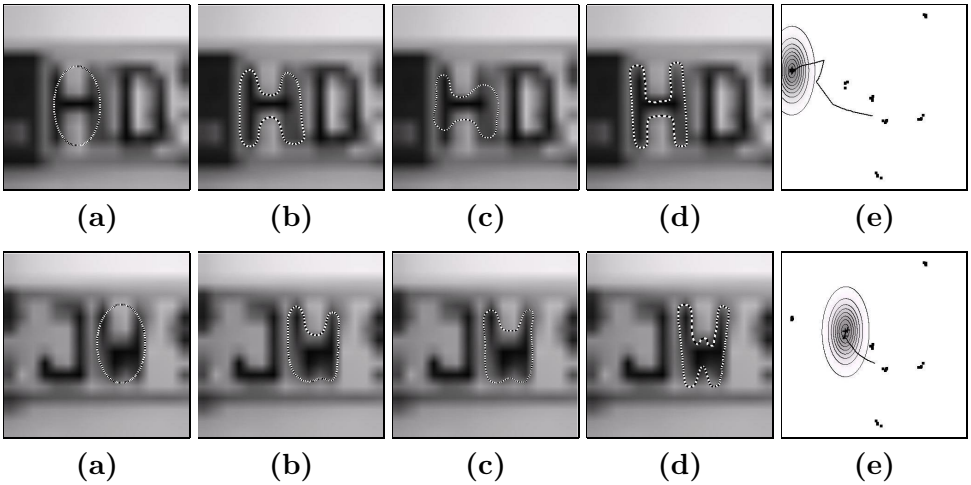


Figure 4.15: Initial contour **(a)**, final segmentation without prior **(b)**, segmentation upon introduction of the linear prior **(c)**, and final segmentation with the nonlinear prior **(d)**. Appropriate projections of the contour evolution with nonlinear prior into the space of contours show the convergence of the contour towards one of the learnt letters **(e)**.

of the training contours.[6]

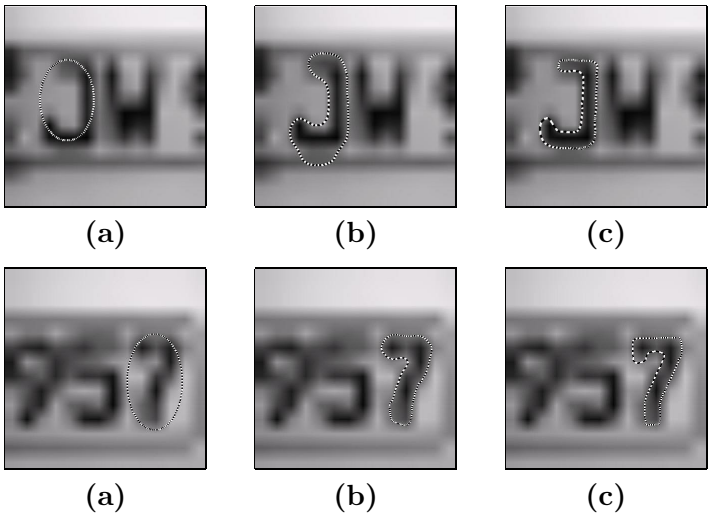


Figure 4.16: Initial contour **(a)**, final segmentation without prior **(b)**, and final segmentation upon introduction of the nonlinear prior **(c)**. With a single nonlinear prior, a number of fairly different shapes can be reconstructed from the subsampled and smoothed input image.

Figure 4.16 shows results of the segmentation approach with the *same* non-

[6]For better visibility, the projection planes were shifted along the third principal component, so as to intersect with the cluster of interest.

linear shape prior, applied to two more shapes. Again, the nonlinear shape prior improves the segmentation results. This demonstrates that one can encode information on a set of fairly different shapes into a single shape prior.

4.7.3 Generalization to Novel Views

Essentially, in all of the above examples, the nonlinear shape prior merely permitted a reconstruction of the training shapes. Although our approach is far more elaborate, one could argue that for the above tasks a simple template matching approach would be sufficient.

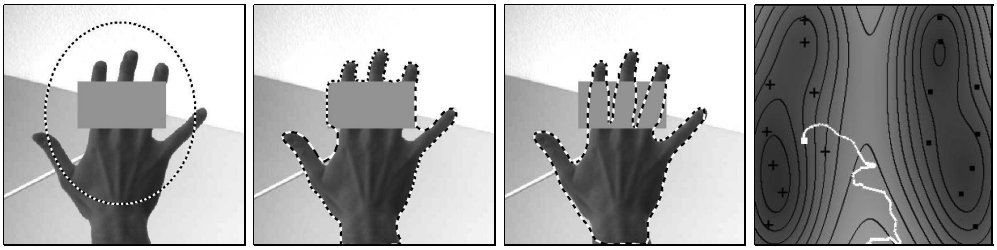
The power of the proposed shape prior lies in the fact that not only it can encode several very different shapes, but also that the prior is a *statistical* prior. This means that it has the capacity to generalize and abstract from the fixed set of training shapes. The consequence is that — as in the case of the linear prior — the respective segmentation process with the nonlinear prior is able to segment novel views of an object which were not present in the training set. This aspect of the nonlinear statistical shape prior will be demonstrated in the following examples.

The training set consists of 9 right hands and 9 left hands, which are shown in Figure 4.7. Figure 4.17 shows the results of a segmentation process without and with the nonlinear shape prior. Due to the nonlinear prior the occlusion is ignored and the silhouette of the hand is reconstructed in areas where it is occluded. The last image shows the training shapes and the estimated shape energy (4.18) in a projection onto the first two principal components (of a linear PCA). Compared to the linear shape energy, shown in Figure 4.7, second image, one can clearly distinguish the two valleys corresponding to the right and left hands. Moreover, the path of the segmenting contour is projected (as a white line) into the density plot. The final segmentation — indicated by a white box — is clearly different from all the training shapes (black crosses). The nonlinear prior does not simply pull the segmenting contour towards one of the training shapes. Instead it pulls the contour towards the valleys of the estimated distribution, i.e. the shaded areas in Figure 4.17, right side. This clearly demonstrates the *statistical nature* of the proposed shape prior.

Similar results can be obtained for the segmentation of different views of the two 3D objects presented in Figure 4.8. Given a set of training views of these two objects, we determined the nonlinear shape energy plotted in Figure 4.8, right side. Figure 4.18 shows results obtained by a gradient descent on the energy (4.22), first with no prior knowledge ($\alpha = 0$), and then with the nonlinear prior ($\alpha > 0$). A comparison shows that for views of both objects, the *same* prior permits to deal with the background clutter, which — in the absense of a prior — is included in the segmentation.

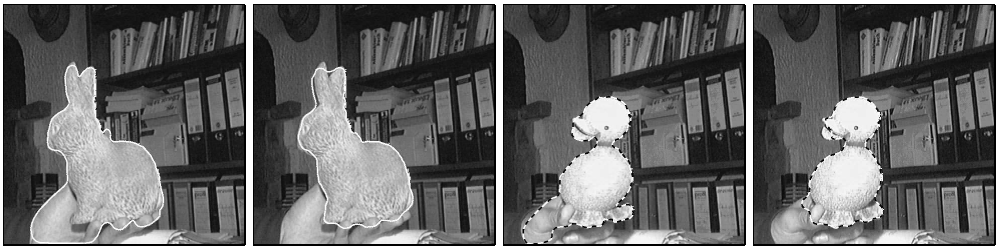
4.7.4 Tracking 3D Objects with Changing Viewpoint

The previous example demonstrated that it is possible to encode the appearance of a given 3D object in terms of the silhouettes associated with different 2D projections. However, in the results of Figure 4.18, one can see small errors



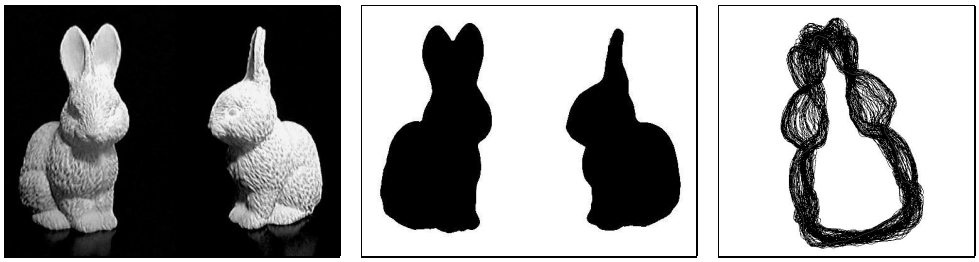
Initial Contour | Segmentation without prior | Segmentation with prior | Projected density estimate

Figure 4.17: Generalization to novel views. Segmentation of a partially occluded hand. The training set contains 9 right and 9 left hand shapes. The training shapes and the estimated energy are shown in a projection onto the first two eigenmodes of a linear PCA (right). The white line indicates the path of the evolving contour upon introduction of the nonlinear shape prior. Note that the final contour does not correspond to any of the training shapes.



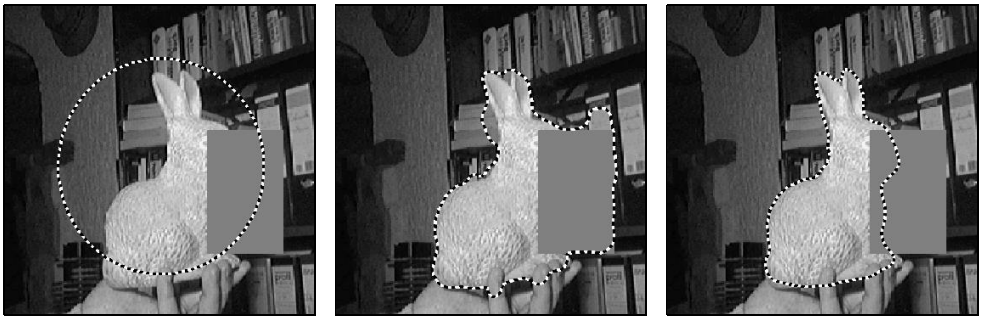
Without prior | With prior | Without prior | With prior

Figure 4.18: Encoding several 3D objects in a single prior. The training set contains several views of the two objects. It is shown together with the estimated energy density in Figure 4.8. The segmentation results show that the nonlinear prior permits to suppress most of the clutter which is included if no prior is used.



2 views of a rabbit | binarized | aligned contours

Figure 4.19: Example views and binarization used for estimating the shape density.



Initial contour No prior With prior

Figure 4.20: Begin of the tracking sequence. Initial contour, segmentation without prior, and segmentation upon introduction of the nonlinear prior on the segmenting contour.

of the final segmentation with prior (around the ears of the rabbit, for example). They indicate that the number of training shapes may not have been sufficiently large for this application. The next example will indeed confirm this presumption.

We will apply the nonlinear shape statistics in an example of tracking an object in 3D with a prior constructed from a large set of 2D views. We binarized 100 views of a rabbit — two of them and the respective binarizations are shown in Figure 4.19. For each of the 100 views we automatically extracted the contours and aligned them with respect to translation, rotation, scaling and cyclic reparameterization of the control points — see Figure 4.19, right side. We calculated the density estimate (4.15), which generates the nonlinear shape prior in equation (4.22).

In a film sequence we moved and rotated the rabbit in front of a cluttered background. Moreover, we artificially introduced an occlusion afterwards. We segmented the first image by the simplified diffusion snake model until convergence, before the shape prior was introduced. The initial contour and the segmentations without and with prior are shown in Figure 4.20. Afterwards we iterated 15 steps in the gradient descent on the full energy for each frame in the sequence.[7] Some sample screen shots of the sequence are shown in Figure 4.21. Note that the viewpoint changes continuously during the sequence.

The training silhouettes and the estimated shape energy are shown in two different 2D projections in Figure 4.22. The path of the evolving contour during the entire sequence corresponds to the white curve. In this projection, the curve

[7]The gradient of the shape prior in (4.23) has a complexity of $O(rmn)$, where n is the number of control points, m is the number of training silhouettes and r is the eigenvalue cutoff. For input images of 83 kpixels and m=100, we measured an average runtime per iteration step of 96 ms for the prior, and 11 ms for the cartoon motion on a 1.2 GHz AMD Athlon. This permitted to do 6 iterations per second. Note, however, that the relative importance of the cartoon motion increases with the size of the image: For an image of 307 kpixels the cartoon motion took 100 ms per step. Note, however, that we did not put much effort into runtime optimization.

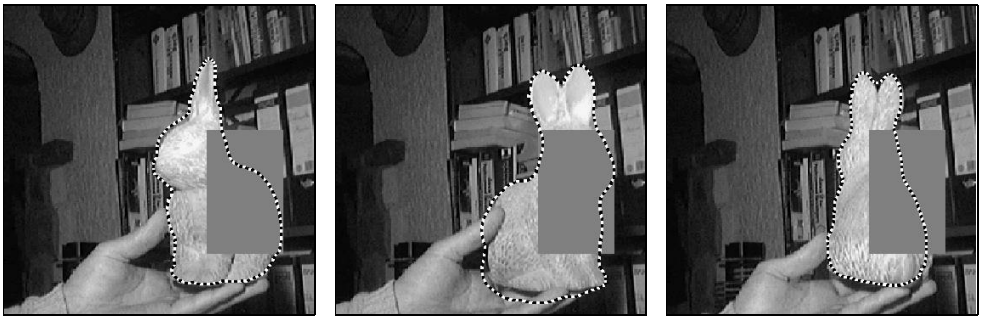
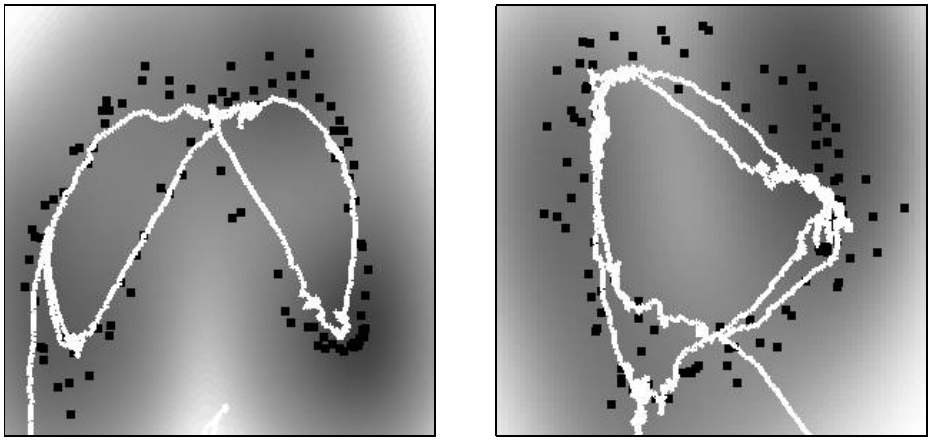


Figure 4.21: Sample screen shots from the tracking sequence.

follows the distribution of training data well, interpolating in areas where no training silhouettes are present. Note that the intersection of the white curve in the center of Figure 4.22, left side, is only due to the projection on 2D. The results show that — given sufficient training data — the shape prior is able to capture fine details such as the ear positions of the rabbit in the various views. Moreover, it generalizes well to novel views not included in the training set and permits a reconstruction of the occluded section throughout the entire sequence.



Projection onto 1st and 2nd principal component

Projection onto 2nd and 4th principal component

Figure 4.22: Tracking sequence visualized. Training data (•), estimated energy density and the contour evolution (white curve) in appropriate 2D projections. The evolving contour — see Figures 4.20 and 4.21 — is restricted to the valleys of low energy induced by the training data.

4.8 Concluding Remarks

The previous examples showed that one can model the distribution of a set of training shapes, after an appropriate nonlinear transformation ϕ, by a Gaussian density in the feature space Y. Although conceptually this may appear to be a minor modification of the original Gaussian model presented in Section 3.3, the strong nonlinearity is able to generate a fundamentally different shape energy. It permits to model multimodal distributions of essentially arbitrary shape, thus generalizing the simple hyperellipsoid associated with the multivariate Gaussian distribution. Compared to the model of mixtures of Gaussians, it is not limited to a sum of hyperellipsoids. Moreover, no prior clustering or classification of the different training shapes is necessary. Rather than specifying explicitly the number of clusters (as generally done in the case of the mixture model), a single granularity parameter, given by the kernel width σ, induces a spatial scale and thereby implicitly determines the number of clusters for a given training set. In several segmentation tasks, we were able to confirm that the resulting nonlinear shape prior can simultaneously encode the silhouettes corresponding to several objects, or the appearance of a real world 3D object in terms of its various 2D projections. Combined in a variational segmentation approach, it can capture even small details of shape variation without mixing different views. Moreover, it is a *statistical* prior in the sense that it permits the segmentation of views of an object which were not part of the training set. Appropriate 2D projections demonstrate how the evolving contour is drawn to the valleys of the statistical distribution induced by the training shapes.

From a Bayesian perspective, the crucial task underlying the construction of a shape prior is to optimally estimate the probability density function from a limited number of sample shapes. Therefore it may be of interest to study shape priors based on other methods of density estimation such as the *Parzen estimator* [151, 144]. Moreover, a more general investigation of the relation between such distances in feature space as the one proposed in equation (4.15) and classical methods of density estimation seems to be promising. Although by far not exhaustively, we will investigate this latter question in Appendix C.

Chapter 5

Shape Statistics in Motion Segmentation

In the present chapter, we will extend the Mumford-Shah model to the problem of motion segmentation. In particular, we will present a spline based implementation which segments two consecutive frames of an image sequence not with respect to the image intensity, but rather with respect to the motion information. Essentially the variational approach consists in approximating the true image motion by a model of piecewise homogeneous motion, where homogeneity is defined in terms of parametric motion models for each of the segmented regions. We will show that in analogy to the grey value case, one can extend the variational approach by a statistical shape prior which measures the dissimilarity of the segmenting contour with respect to a set of training contours. The resulting segmentation process is derived by gradient descent on a *single* energy functional, which simultaneously updates the parametric motion models in the separate regions and the position of the motion discontinuity represented by the contour. Combined with the shape prior, the variational approach deforms the contour in such a way as to maximize both the homogeneity of motion in each region, and the similarity of the contour with respect to a statistically encoded set of training shapes.

5.1 Introduction and Related Work

Based on the Mumford-Shah model and its cartoon limit, we presented in Section 2.4 two spline-based segmentation approaches, namely the diffusion snake (DS) and the simplified diffusion snake (SDS), which evolve a spline contour so as to maximize the homogeneity with respect to the grey values in each segmented region. While the DS measures the homogeneity by approximating with a function of piecewise smooth grey value, the SDS approximates the image with a function of piecewise constant grey value.

However, in many real world scenarios, the object of interest may not be easily discriminated from the background by its intensity appearance. Then a segmentation approach which is purely based on the grey value information (of a single frame) may fail, as shown in Figure 5.1. In particular, for biological

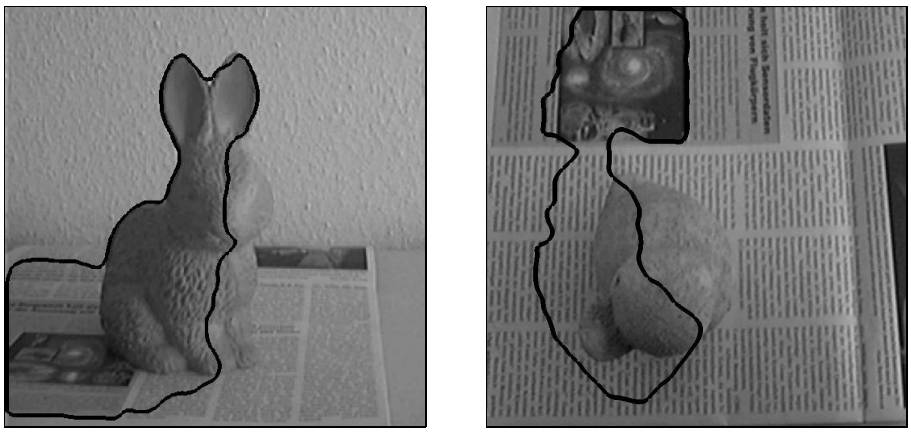


Figure 5.1: Segmentations by image *intensity* in difficult lighting conditions. Due to shadows and similar grey value information of object and background, the segmentation by piecewise constant grey value with the simplified diffusion snake fails to capture the object of interest — in this case the rabbit or the duck.

vision systems such as the human vision, the motion information of an object is an important cue. In the following we will therefore extend the homogeneity measure of the Mumford-Shah functional to motion. In particular, we will present two modifications of the diffusion snake which approximate the motion information contained in an image sequence (given in terms of two consecutive images) by a model of *piecewise constant* or *piecewise affine* flow fields.

Discontinuity-preserving motion estimation by variational models and related partial differential equations have a long tradition in computer vision. In some approaches, the motion discontinuities are modeled implicitly in terms of appropriate (non-quadratic) regularizers [137, 158, 15, 128, 113, 188]. Other approaches pursue separate steps of variational motion estimation on disjoint sets with a shape optimization procedure [157, 160, 73, 141].

For the case of grey value segmentation, there exist some region-based variational approaches with explicit discontinuities, as discussed in Chapter 2, and extensions to color and texture segmentation [204]. The Mumford-Shah functional (2.11) has been adapted to the problem of motion segmentation in [137], however there the author again prefers an implicit model of the discontinuity by reverting to approximations in terms of Γ-convergence as studied in [7].

In contrast, the following variational approach to motion segmentation is based on an *explicit* contour description. It has several favorable properties: Firstly, the gradient descent on a *single* energy functional jointly solves the problems of segmentation and estimation of piecewise affine motion fields. Moreover, as in the case of grey value segmentation, the explicit representation of the contour permits to incorporate a statistical prior on the shape of the segmenting contour.

Prior knowledge in terms of motion models was incorporated in motion estimation and motion segmentation by [140, 141]. In contrast to this approach,

we focus on prior knowledge with respect to shape and thus directly address the problem of determining accurate motion boundaries in a generative way. Deformable shape models were combined with motion segmentation in [107]. However, there the authors did not propose a variational integration of motion segmentation and shape prior. Rather they optimize a small number of shape parameters by simulated annealing, which — unlike our approach — cannot be applied to more general shape priors (such as the contour length).

5.2 Variational Motion Segmentation

Let $f(x,t)$ be an image sequence which is assumed to be differentiable. We assume moreover that the intensity of a moving point is constant throughout time. Then we obtain a continuity equation given by the classical *optic flow constraint*:

$$\frac{d}{dt}f(x,t) = \frac{\partial}{\partial t}f + w^t \nabla f = 0, \tag{5.1}$$

where $w = \frac{dx}{dt}$ denotes the local velocity. Given two consecutive images f_1 and f_2 from this sequence, we can approximate[1] $\frac{\partial}{\partial t}f \approx (f_2 - f_1)$ and $\nabla f \approx \frac{1}{2}\nabla(f_1 + f_2)$.

We propose to segment the image plane into areas R_i of parametric motion $w_i = w_i(\xi_i)$ by minimizing the energy functional

$$E(\xi, C) = \sum_i \int_{R_i} \left(f_2 - f_1 + \frac{w_i^t}{2}\nabla(f_1 + f_2) \right)^2 dx + \nu\, E_c(C) \tag{5.2}$$

simultaneously with respect to both the contour C, which separates the regions R_i, and the parameters $\xi = \{\xi_i\}$ which define the motion in region R_i. Possible motion models will be detailed in Section 5.3. The term E_c represents an *internal shape energy*, such as the length of the contour or a more elaborate shape dissimilarity measure, as detailed in the previous chapters.

With the extended velocity vector $v = \binom{w}{1}$ and the spatio-temporal structure tensor [14]

$$S = (\nabla_3 f)(\nabla_3 f)^t, \quad \text{with } \nabla_3 f = \begin{pmatrix} \nabla f \\ \frac{\partial}{\partial t} f \end{pmatrix},$$

the energy (5.2) can be rewritten as

$$E(\xi, C) = \sum_i \int_{R_i} \left(v_i^t\, S\, v_i \right) dx \; + \; \nu\, E_c(C). \tag{5.3}$$

In practice, the homogeneity term shows a bias towards velocity vectors of large magnitude. As proposed in [72], we therefore perform an isotropy compensation of the structure tensor by replacing S with $S - \lambda_3\, I$, where λ_3 is the smallest eigenvalue of S, and I is the 3×3 unit matrix.

[1] The discretization of spatial and temporal derivatives implys that velocities are measured in pixels per frame.

5.3 Piecewise Homogeneous Motion

The proposed *motion energy* (5.3) can be interpreted as an extension of the Mumford-Shah model (2.11) to the problem of motion segmentation. Rather than measuring the grey value homogeneity, it measures the homogeneity with respect to parametric motion models in the respective regions. In the following we will focus on the two cases of constant motion (2 parameters) and affine motion (6 parameters). However, depending on the application other motion models can also be used, provided that the extended velocity vector is linear in the parameters. Examples are a 4-parameter model to describe Euclidean transformations (translation, rotation and scaling) or an 8-parameter model, which (in contrast to the affine one) permits to model rigid motion under *perspective projection* (rather than orthographic projection).

However, a larger number of parameters does not neccessarily improve the proposed method. Although a higher degree of freedom for the estimated motion fields permits more flexibility for modeling the image motion, the functional (5.6) can expected to also have more local minima, which poses problems to a local minimization scheme. Essentially the local minima are due to the fact that the motion of a given image patch may be explained by several, rather different choices of parameters (if the number of parameters is large). In several applications, we were able to confirm this difficulty.

For the model of **piecewise constant motion**, the extended velocity vector for region R_i is given by:

$$v_i = T\xi_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} (a_i, b_i, 1)^t, \tag{5.4}$$

where a_i and b_i denote the velocity in x- and y-direction. For the model of **piecewise affine motion**, the velocity at a point (x, y) is given by:

$$v_i = T\xi_i = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} (a_i, b_i, c_i, d_i, e_i, f_i, 1)^t, \tag{5.5}$$

with 6 parameters defining an affine motion in region R_i.

Inserting these parametric motion models into the motion energy (5.3), we get:

$$E(\xi, C) = \sum_i \xi_i^t \, Q_i \, \xi_i \; + \; \nu \, E_c(C), \tag{5.6}$$

where

$$Q_i = \int_{R_i} T^t S T \, dx = \begin{pmatrix} \bar{Q}_i & q_i \\ q_i^t & \gamma_i \end{pmatrix}.$$

Depending on the model, the submatrix $\bar{Q}_i$ and the vector q_i have the dimension 2 for the constant motion model and 6 for the affine model, respectively.

5.4 Motion Competition

The motion energy (5.6) has to be simultaneously minimized both with respect to the evolving contour and with respect to the motion parameters $\{\bar{\xi}_i\}$, where $\xi_i = \binom{\bar{\xi}_i}{1}$ is defined with respect to the chosen motion model — see equations (5.4) and (5.5).

Minimization with respect to $\bar{\xi}_i$ results in the linear equation:

$$\bar{Q}_i \, \bar{\xi}_i = -q_i.$$

Due to the well-known *aperture problem*, the symmetric square matrix $\bar{Q}_i$ may not be invertible. In this case, we need to impose an additional constraint. Choosing the solution $\bar{\xi}_i$ of minimal length, amounts to applying the pseudo-inverse $\bar{Q}_i^\dagger$ (cf. [72]):

$$\bar{\xi}_i = -\bar{Q}_i^\dagger \, q_i. \tag{5.7}$$

Using Green's theorem, minimization of (5.6) with respect to the contour C results in the evolution equation (cf. Section 2.5):

$$\frac{dC}{dt} = -\frac{dE}{dC} = \left(e^- - e^+ \right) \boldsymbol{n} - \nu \, \frac{dE_c}{dC}. \tag{5.8}$$

The last term minimizes the internal shape energy which will be treated in the next section. The superscripts $j = +/-$ denote the two regions to the left and to the right of the respective contour point (in the sense of the contour parameterization), and $\boldsymbol{n}$ is the normal on the contour pointing out of the region R_+.

The adjacent regions compete for the contour in terms of the associated energy densities[2]

$$e^j = v_j^t \, S \, v_j. \tag{5.9}$$

This *motion competition* enforces regions of homogeneous optic flow, thus separating regions moving at different velocities w_j.

For comparing different motion hypotheses, it is suggested in [72] to normalize the cost function in (5.9) by replacing

$$e^j = v_j^t \, S \, v_j \qquad \text{with} \qquad \frac{v_j^t \, S \, v_j}{||v_j||^2 \, \mathrm{tr} S}$$

in the evolution equation (5.8).[3]

5.5 Contour Evolution

As in the previous chapters, we will implement the motion competition algorithm by an explicitly represented contour:

$$C : [0,1] \to \Omega, \quad C(s) = \sum_{n=1}^{N} p_n \, B_n(s), \tag{5.10}$$

[2] In the equivalent probabilistic interpretation, this energy density represents the *log-likelihood* of the probability that a given location is part of one or the other motion region.

[3] Although this modification is not strictly derived by minimizing energy (5.6), it tends to slightly improve the contour evolution.

with spline control points $p_n = (x_n\,,\, y_n)^t$ and periodic quadratic B-spline basis functions B_n. This permits a relatively fast numerical optimization. Moreover, it facilitates the incorporation of a statistical shape prior on the control point vector $z = (x_1, y_1, \ldots, x_N, y_N)^t$, as explained in Chapters 3 and 4.

For the internal shape energy E_c, we will use the contour length measure (2.21) and the similarity invariant shape prior (3.23) with the linear Gaussian model (3.17). As in the case of grey value segmentation, the motion segmentation can also be combined with a nonlinear shape prior based on the Gaussian in feature space, as introduced in Section 4.4. For the time being, we have not done this.

As in the case of the corresponding grey value model, the curve evolution (5.8) can be converted to an evolution equation for the spline control points by inserting the definition (5.10) of the contour as a spline curve. The equation is discretized with a set of nodes s_i along the contour, where s_i is chosen as the point where the respective spline basis function B_i attains its maximum. Including the contribution of the internal shape energy, we obtain for the control point (x_m, y_m):

$$\begin{aligned} \frac{dx_m(t)}{dt} &= \sum_{k=1}^{N} \left(\mathbf{B}^{-1}\right)_{mk} \left(e^+(s_k,t) - e^-(s_k,t)\right) n_x - \nu \left(\frac{dE_c}{dz}\right)_{2m-1}, \\ \frac{dy_m(t)}{dt} &= \sum_{k=1}^{N} \left(\mathbf{B}^{-1}\right)_{mk} \left(e^+(s_k,t) - e^-(s_k,t)\right) n_y - \nu \left(\frac{dE_c}{dz}\right)_{2m}, \end{aligned} \tag{5.11}$$

where n_x and n_y denotes the x- and y-coordinates of the normal vector, and the indices $2m-1$ and $2m$ refer to the components of the given vector z which are associated with the control point (x_m, y_m). The cyclic tridiagonal matrix $\mathbf{B}$ contains the spline basis functions evaluated at the nodes: $B_{ij} = B_i(s_j)$.

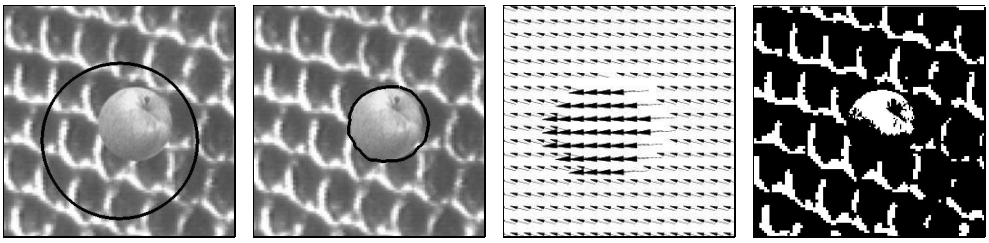
The two terms in (5.11) can be interpreted as follows:

- The first term forces the contour towards the boundaries of the homogeneous motion fields by minimizing the motion inhomogeneity in the adjoining regions, measured in terms of the energy density (5.9).
- The last term minimizes the internal shape energy — in our case the length of the contour (2.21), a shape dissimilarity measure of the form (3.23), or a linear combination of both.

5.6 Experimental Results

Given two consecutive images of a motion sequence[4], we minimize the total energy (5.6) by iterating the contour evolution equation (5.11) in alternation with an update of the motion estimation (5.7) in the adjoining regions.

[4] All figures will only show the first of the two consecutive images. With displacements of at most 5 pixels for an image of size 256^2, the differences between the two frames are almost imperceivable if they are put next to eachother.



Initial contour | Final contour | Estimated flow field (closeup) | Binarized image

Figure 5.2: Initial and final contour obtained by gradient descent on the functional (5.6) with the model (5.4) of piecewise constant motion for a moving apple on a differently moving background. A zoom of the estimated motion field corresponding to the final contour shows that the two separated motion fields are fairly similar. A binarization of the first input image indicates that a segmentation of the apple based on the hypothesis of piecewise constant intensity would fail.

5.6.1 Intensity-based versus Motion-based Segmentation

The first example in Figure 5.2 shows an artificial sequence of an apple which is translated, with the background translated at a different velocity and in a different direction. This can be considered a simplified synthetic analogue with the case of a moving object and a moving camera. The Initial and the final contour show how the two differently moving regions are separated during the minimization. The final flow field estimation shows the two different motion fields which were estimated.

Although inspired by a grey value segmentation approach, the proposed motion segmentation is substantially different from grey value segmentation in that it segments the image plane into regions of constant motion rather than constant grey value. Segmenting the previous example of the apple sequence based on the hypothesis of piecewise constant grey value would entirely fail as can be seen from the corresponding binarized image in Figure 5.2, last image: About half of the apple has disappeared although the background structure is still quite prominent.

The completely different properties of grey value and motion segmentation are also demonstrated on the example in Figure 5.3, where the rabbit is moving with respect to the background. Due to the difficult lighting conditions in this example, the image grey value is not a good cue for segmentation and therefore segmentation based on grey value constancy fails — see Figure 5.3, bottom right image. The segmentation based on motion constancy (with the same initialization) gives a better result, as portrayed by a number of intermediate steps in Figure 5.3, which were taken during the gradient descent sequence on functional (5.6) with the model (5.4) of piecewise constant motion.

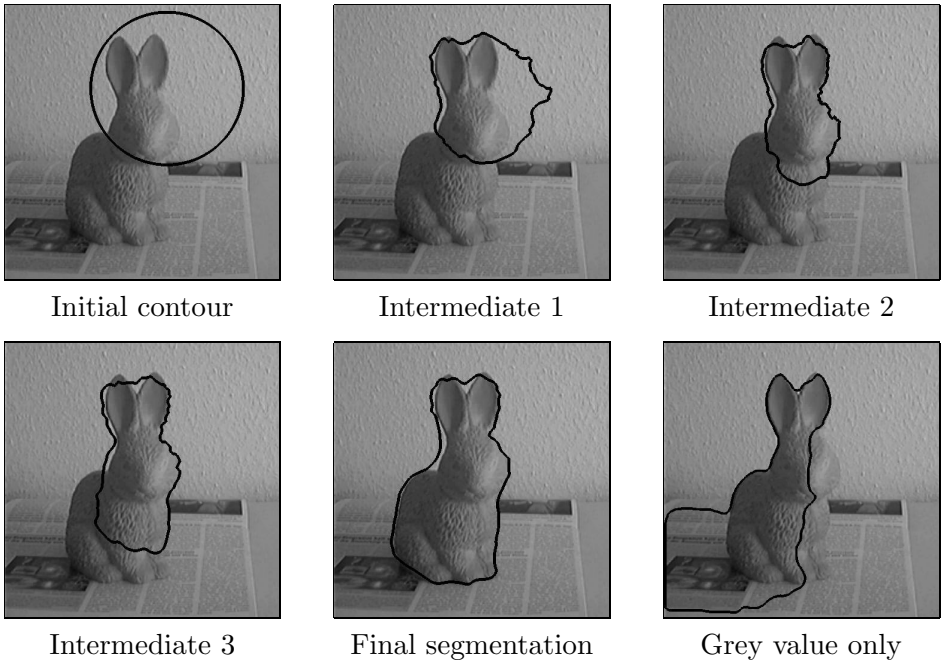


Figure 5.3: Motion segmentation versus grey value segmentation. Contour evolution for model (5.6) of piecewise constant motion (5.4) and final contour for model (2.24) of piecewise constant intensity (bottom right). Due to the difficult lighting conditions, the image intensity is not a reliable cue for segmentation.

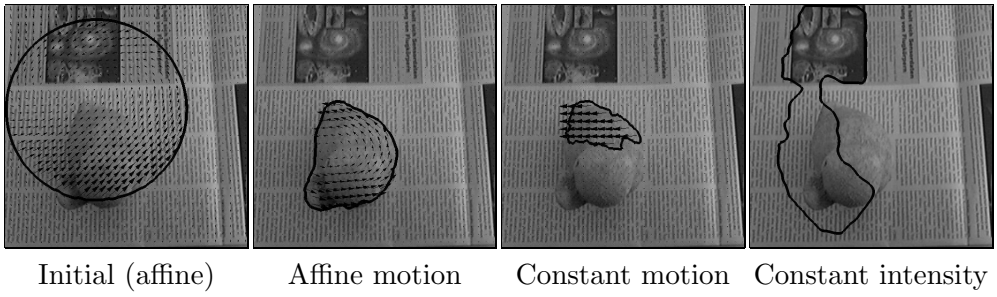


Figure 5.4: Model comparison. Initial contour and final segmentations obtained by gradient descent on the functional (5.6) for the models of piecewise affine motion (5.5), piecewise constant motion (5.4), and for the corresponding model of piecewise constant intensity. The input images show a duck figure rotated on a newspaper. Note that the affine motion model captures the rotation and thereby correctly segments the duck, whereas the model of constant motion only captures those parts which show approximately constant motion. The segmentation by intensity is completely misled by background clutter and the difficult lighting conditions.

5.6.2 Piecewise Constant versus Piecewise Affine Motion

The previous examples showed, that in cases where the object motion is different from the background motion, a segmentation by piecewise constant motion is successful, even though the object is not easily discriminated from the background by its appearance.

Yet, as in the case of piecewise constant grey value, segmentation by piecewise constant motion can only be successful as long as the corresponding hypothesis applies to the given image (sequence). If instead the object is rotating or moving towards the camera such that the motion field is divergent, then the assumption of piecewise constant motion is violated and the corresponding segmentation approach will fail. In this case, a model based on piecewise affine motion should be more successful.[5] This is demonstrated in Figure 5.4. The figure of the duck is rotating on a static newspaper. Even for the human eye the duck is hardly discernible, due to similar intensities of object and background and the lighting conditions. For the same initial contour (first image), we performed a gradient descent on the functional (5.6) for piecewise affine motion (5.5) and piecewise constant motion (5.4), and on the functional (2.24) for a segmentation by piecewise constant intensity. For the motion segmentation the respective contour and the estimated flow field were superimposed on the image. The segmentation by piecewise affine motion not only successfully segments the object, but also captures the rotatory motion and correctly determines the center of rotation. In contrast, the model of piecewise constant motion only segments a part of the object which complies with the assumption of constant motion. The segmentation by constant intensity is completely misled by the background clutter and effects of shading, as shown by the last image of Figure 5.4.

5.6.3 Convergence over Large Distances

The examples of the rabbit in Figure 5.3 and the duck in Figure 5.4 show a central property of our approach: Since it is a region-based approach, the contour tends to converge over fairly large distances. This aspect is highlighted by the example of a moving bus in an otherwise static scene in Figure 5.5.

During the gradient descent minimization both the contour and the estimate of the flow field are improved simultaneously. The flow fields estimated for the initial and the final contour are shown in the last two images of Figure 5.5. Note that the final estimate of the motion of the bus is strongly improved compared to the initial one.

5.6.4 Moving Background

A central difficulty in motion estimation is the case of separating differently moving regions. In practical applications, this problem arises if the camera itself is moving. Commonly [140, 15] the camera motion is eliminated by determining the dominant motion in a robust estimation framework [98] first and

[5] Apart from scaling, rotation and translation, the affine model also encompasses shearing, which permits to model 3D rotation of planar objects under orthographic projection.

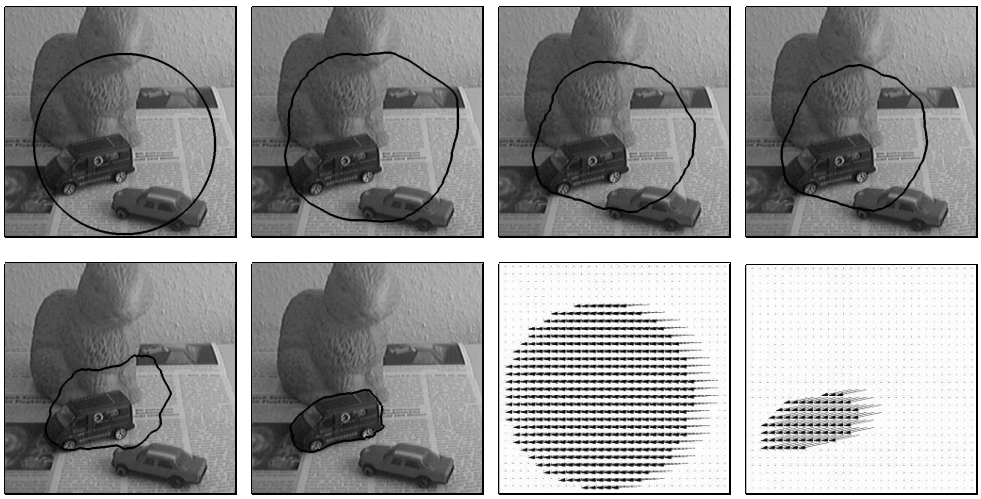


Figure 5.5: Convergence over large distances. Contour evolution for two images of a moving bus. The estimated flow field corresponding to the initial and final contour are shown in a close-up. The estimated object motion is gradually improved during the contour evolution.

then subtracting the latter. However, the assumption that the moving background fills the dominant part of the image plane may not always be valid.[6]

The variational approach (5.6) does not rely on any assumptions about the size of the segmented motion fields. In fact, examples such as the moving bus sequence in Figure 5.5 show that the object motion does not even have to fill the dominant part of the initially enclosed area for the minimization to converge correctly. This property is due to the fact that both the contour evolution (5.8) and the motion estimation (5.7) were derived by minimizing a *single* energy functional. It is in fact the *non-robust* estimation of the motion inside and outside the contour which defines the driving force for the contour via the energy densities e^+ and e^- in the evolution equation (5.8). In the example in Figure 5.5, a robust estimation of the motion inside the initial contour would, for example, estimate a zero velocity and the contour would not evolve towards the bus.

The following examples show that similar convergence properties of our method can be observed if both the object and the background are moving. Figure 5.6 shows a snapshot of a segment of wallpaper in which a circular area in the center was artificially rotated in one sense and the background in the opposite sense, as shown by the ground truth motion field in Figure 5.6, bottom right. The initial contour was placed in a location where less than half of its inside area overlapped the motion inside the circular area. The contour evolution indicates how the two affine motion fields are progressively separated during the energy minimization. Not only does the final segmentation match

[6] In [15], for example, it is stated that the robust estimation of the background motion works well on an artificial sequence (involving translatory motion only) if the background motion takes up at least 60% of the image plane.

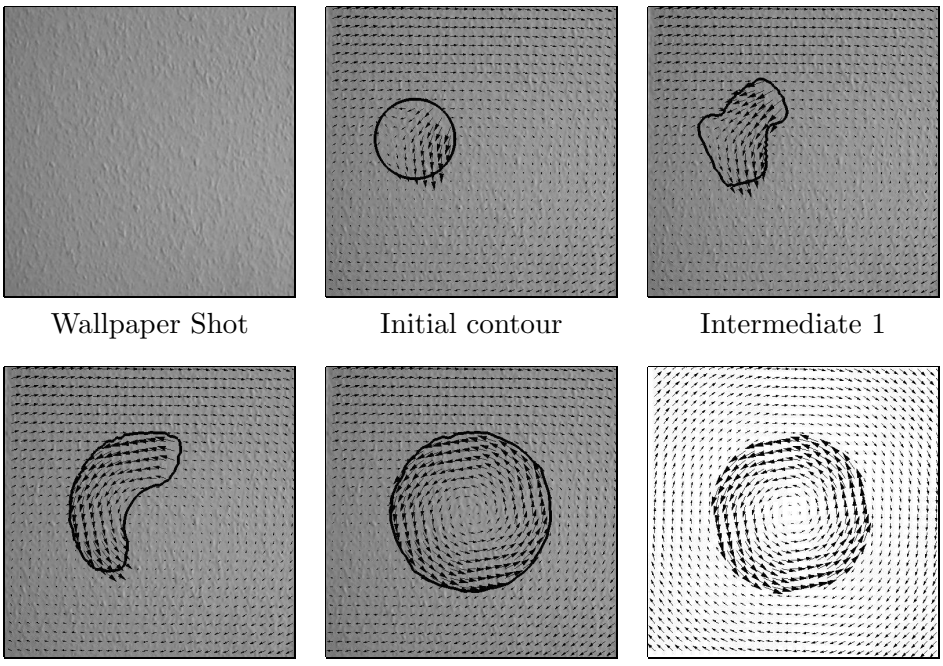


Figure 5.6: Separating two affine motion fields by gradient descent on the functional (5.6) for the piecewise affine motion model (5.5). The input sequence shows a wallpaper with a circular area in the center rotated in one sense, and the background rotated in the opposite sense, as indicated by the true motion field on the bottom right. During energy minimization the estimated motion fields are continuously improved, the two motion fields are separated, and the circular area is correctly segmented. Note that the circular area cannot be detected based on grey value information.

the rotated area exactly, but also the estimated motion reflects the ground truth well. Note that this example demonstrates the fundamental difference between the proposed motion segmentation and the corresponding grey value segmentation approach introduced in Chapter 2: The rotated section of the wallpaper does not differ from the rest of the image by its grey value information — see Figure 5.6, top left.

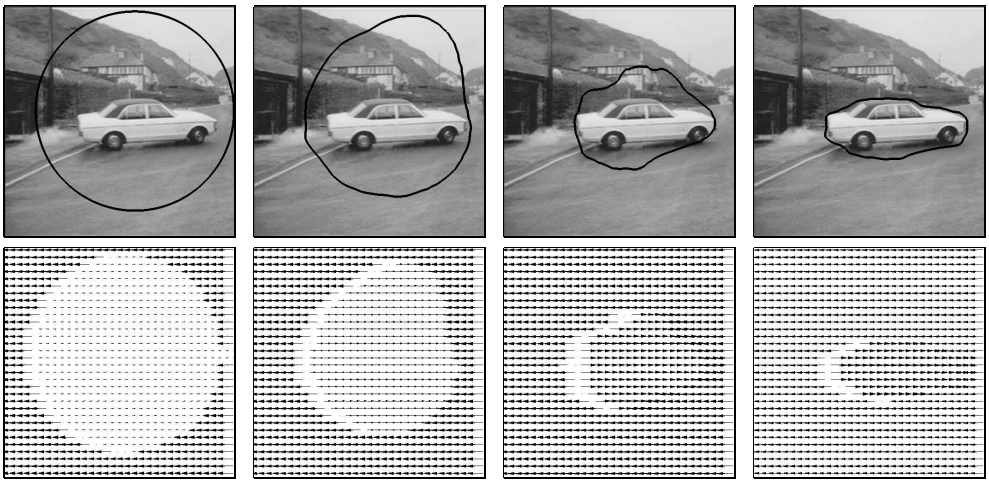


Figure 5.7: Example from the well-known Avengers sequence. Energy minimization for the model of piecewise constant motion for the example of a moving car captured by a moving camera. Despite large and not exclusively translatory motion and little grey value structure of car and street, the final segmentation is rather good. Note that the discrepancy between car and street is due to the shadow moving with the car.

Figure 5.7 shows an example of a moving car from the well-known Avengers sequence.[7] The car performs a more or less translatory motion. Due to the camera motion, the background also performs a motion which (apart from a zoom) is mostly translatory. Several steps in the energy minimization process show, that the contour converges towards the object boundaries and that the two motion fields are progressively separated. Minor discrepancies between the final contour and the object boundaries probably have several reasons: Firstly, the motion hypothesis of piecewise constant motion is only a rough approximation of the true motion. Secondly, the car motion in this sequence is fairly large (around 4 pixels in most areas). And thirdly, both the white car and the grey street have little grey value structure. Discrepancies between car and street are also due to the shadow moving with the car.

5.6.5 Motion Segmentation with a Statistical Shape Prior

In cases of ambiguous motion information, e.g. due to missing or misleading information, the proposed motion segmentation may fail to converge to the correct result. If the object of interest is known, one may solve this problem by introducing some prior knowledge into the segmentation approach.

In the next example, the object of interest is a moving hand, performing a more or less translatory motion. As explained in Sections 3.3 and 3.4, a statistical shape energy was derived from a set of 10 hand shapes, none of which corresponds to the hand in the image sequence.

[7] We thank P. Bouthemy and his group for providing us with the image data from the Avengers sequence.

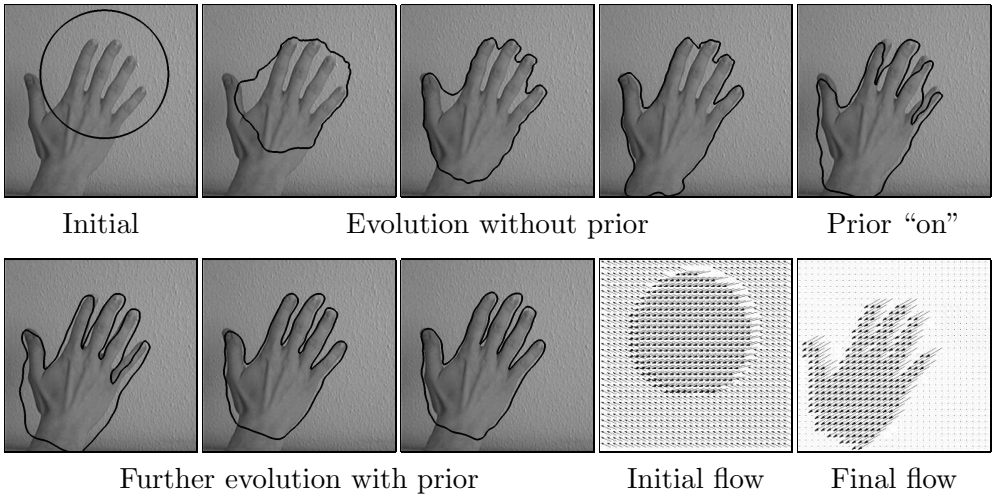


Figure 5.8: Effect of the statistical shape prior for a hand moving to the bottom left. The statistical shape prior is introduced upon stationarity after the fourth frame. Initial and final estimates of the flow field show the improved separation of the two motion fields. The final segmentation is cut at the wrist, because the training shapes were all cut there for simplicity.

We will demonstrate the effect of this shape prior on the motion segmentation process by introducing the shape energy in two different ways.

Switching on the Shape Prior During the Contour Evolution

First we minimize the energy (5.6) without any shape prior ($\alpha = 0$) until stationarity — see Figure 5.8, fourth image. Then we apply the cyclical permutation of spline control points which — given the optimal similarity transformation — best aligns the present contour with the mean of the training contours — see Section 3.1.4. Finally, we switch on the shape prior ($\alpha > 0$) and minimize the total energy (5.6) until convergence — see Figure 5.8, eighth image.

The result shows that the shape prior improves segmentation in areas where the motion information is not strong enough to drive the segmentation process — such as in the area between the fingers.

The estimated flow fields corresponding to the initial and the final contour show that the energy minimization separates the regions corresponding to different motion. During the contour evolution the corresponding motion estimation is gradually improved.

Contour Evolution in the Familiar Subspace

Rather than introducing the shape prior during the evolution, it can be incorporated from the very start. For the same example sequence, a contour evolution with the shape prior is shown in Figure 5.9. The evolution of the estimated, piecewise constant flow field associated with the contour evolution shows that the estimate of the object motion progressively improves during the contour

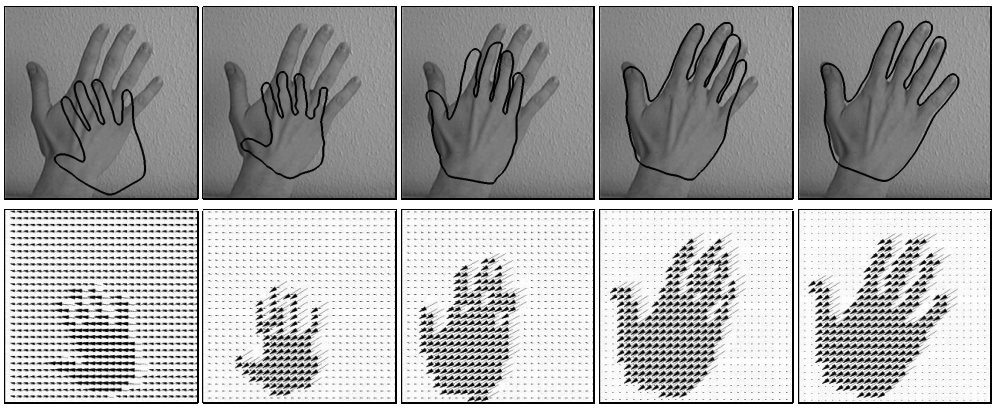


Figure 5.9: Motion segmentation with statistical shape prior. During the contour evolution (top row, from left to right) the motion estimates (bottom) are progressively updated. Compared to the example in Figure 5.8, the shape prior is incorporated from the very start.

evolution — see Figure 5.9, bottom row. In particular, the final segmentation clearly separates the static background from the moving hand shape. Due to the shape prior, the contour is restricted to the subspace of familiar contours throughout the evolution process.

Statistical Shape Prior in Piecewise Affine Motion Segmentation

The previous examples demonstrated the effect of the shape prior on the segmentation process associated with the piecewise constant motion model. Figure 5.10 shows an example of a statistical shape prior favoring hand shapes in a segmentation process with the model (5.5) of piecewise affine motion.

Again the contour is restricted to the submanifold of familiar shapes. It evolves so as to separate differently moving regions while at the same time complying with the prior shape knowledge. The segmenting contour and the estimated piecewise affine flow field are superimposed on one of the two input images. While the initially estimated affine flow fields inside and outside the contour are fairly similar, they are progressively separated during the contour evolution. In the final image, the static background and the rotatory hand motion are well captured.

5.6.6 Dealing with Occlusion

In the above examples of a moving hand, the statistical shape prior improved the convergence towards the desired segmentation — see Figure 5.8.

In a final example, we go one step further and artificially perturb the motion information by partially occluding the moving hand with a static structured object. Figure 5.11 shows the initial and the final contour obtained by minimizing the total energy (5.6) without any shape prior ($\alpha = 0$). Note that the contour separates moving from non-moving regions, given the constraint that no split-

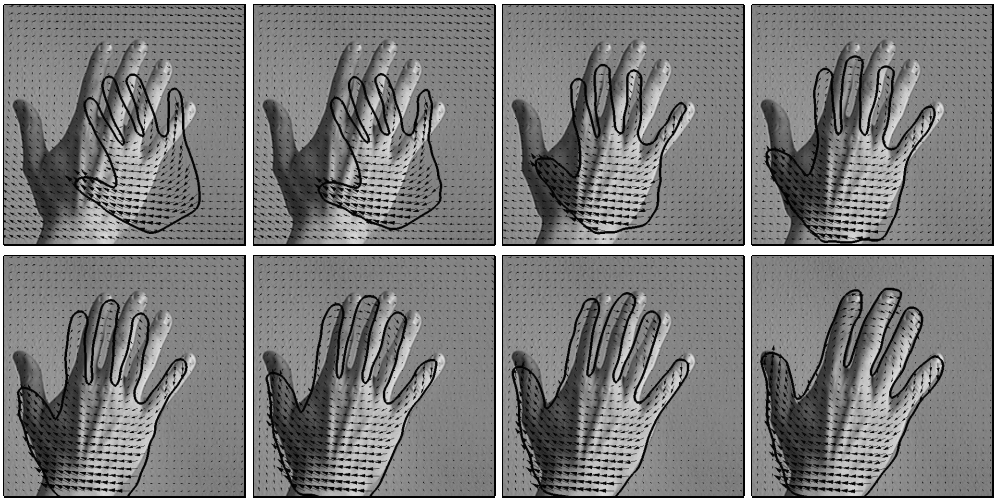


Figure 5.10: Knowledge-driven motion segmentation. Gradient descent evolution for the functional (5.6) with the piecewise affine motion model (5.5) and a statistical prior (3.23) favoring hand shapes. The input sequence is a rotating hand in front of a static background. During the contour evolution the estimated motion fields are continuously improved, while the statistical prior restricts the contour to the submanifold of familiar shapes.

ting of the contour is permitted. For a comparison, the right image shows a similar segmentation obtained with a larger number of 400 control points. It shows that all moving regions are segmented at a higher spatial resolution.

Figure 5.12 shows a contour evolution obtained with a statistical shape prior on the same sequence of a moving hand partly occluded by a static bar. Due to the shape prior, the occlusion is ignored although it is not in accordance

Initial contour Final (100 points) Final (400 points)

Figure 5.11: Motion segmentation without shape prior for a moving hand occluded by a static object. Note that the contour separates moving and non-moving regions. A contour splitting is not permitted. The image on the right shows the final segmentation for a spline curve of 400 rather than 100 control points, which permits a higher spatial resolution.

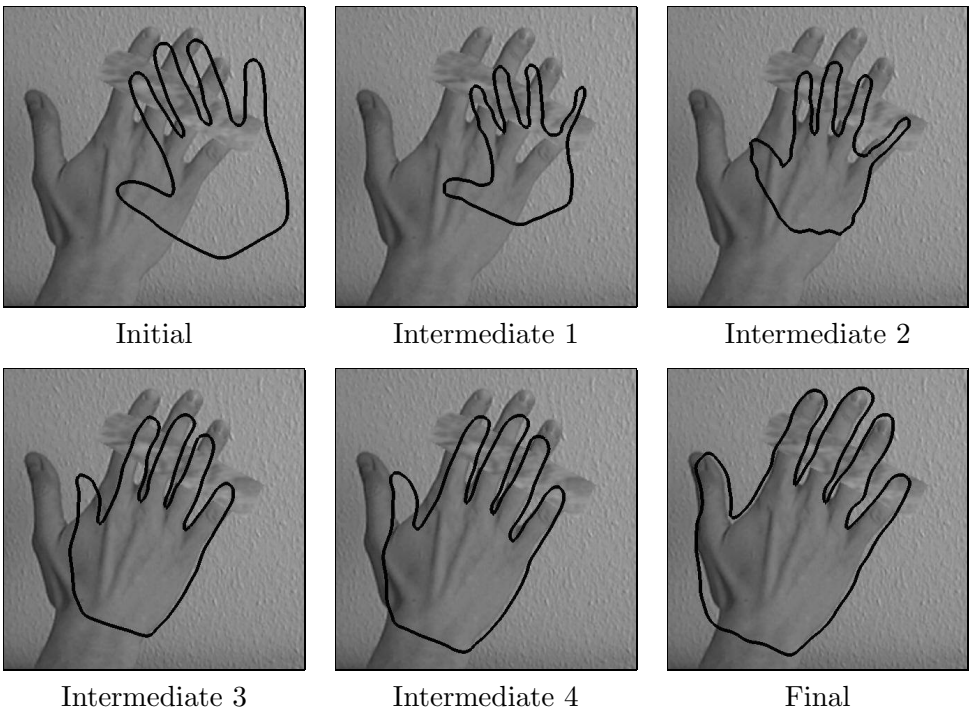


Figure 5.12: Motion segmentation with statistical shape prior for a moving hand occluded by a static object. Note that in this example it appears energetically favorable for the contour to decrease in size during the first iteration steps (2nd and 3rd image). Compared to the segmentation without shape prior in Figure 5.11, center, the effect of the occlusion is compensated by the statistical prior.

with the hand motion. The shape prior permits a reconstruction of the hand silhouette in areas which do not comply with the motion model.

5.7 Concluding Remarks

In this chapter, we presented an extension of the Mumford-Shah model to the problem of motion segmentation. In particular, we detailed an implementation analogous to the diffusion snake which permits to segment the image into piecewise homogeneous motion fields. We presented results for the segmentation of piecewise constant and piecewise affine motion fields. We demonstrated by several examples, that the motion segmentation and the corresponding grey value segmentation are fundamentally different. In particular, objects which cannot be discerned from the background by their appearance can be segmented satisfactorily due to their relative motion.

We proposed two internal shape energies:

- The first one is a purely geometric prior on the length of the contour which is commonly used for modeling elastica such as the classical snake.

It induces a rubber-band like behavior of the segmenting contour and prevents the formation of cusps during the evolution. This shape regularization permits to handle ambiguities in the motion information due to noise. Moreover, it compensates for missing motion information, induced for example by the well-known *aperture problem*, which essentially states that motion information cannot be obtained in directions of constant grey value.

- The second internal energy is a statistical shape energy. Just as in the grey value case, the minimization of this shape dissimilarity measure effectively restricts the evolving motion discontinuity set to a submanifold of familiar shapes. We showed that this facilitates the segmentation of more complex moving shapes. Moreover, it permits to compensate for misleading motion information such that occlusions of the moving object are ignored.

Compared to other approaches, both the evolution of the segmenting contour and the estimation of the affine motion are derived from a *single* energy functional. This means that neither is there any heuristic method to displace the contours, nor do we need to revert to elaborate robust estimators to eliminate the (dominant) motion. The approach involves no prior spatial or temporal smoothing of the image sequence. All derivatives are determined by finite differences. Due to the explicit representation of the contour, the algorithm is fairly fast. For example, the contour evolution for the Avengers sequence shown in Figure 5.7 took less than 15 seconds to converge on a 300 MHz SUN Ultra 10.[8] Real-time implementations are therefore conceivable.

The proposed approach can be extended and improved in several ways:

- Due to the linearization in the optic flow constraint (5.1), the approach cannot cope with large motion. This can be solved by a multiscale implementation for the motion estimates in each region, as done for example in [128].

- For the internal shape energy one can choose more elaborate shape dissimilarity measures such as the one introduced in Chapter 4. As in the case of grey value segmentation this would permit to segment several classes of more complex shapes on the basis of their relative motion.

[8] The evolution for the piecewise *affine* model is somewhat slower, because matrix averaging and matrix inversion are done for a larger matrix (of size 6×6 rather than 2×2). Again, no particular emphasis has been put on runtime optimization.

Chapter 6

Conclusion

6.1 Summary

Variational Combination of External and Internal Information

The central topic of this work is the integration of low-level segmentation cues and high-level shape dissimilarity measures in a variational framework. Segmentations of a given grey value input image f (or image sequence) are obtained by minimizing the total energy

$$E(u, C) = E_{ext}(u, C) \; + \; \alpha E_{int}(C) \tag{6.1}$$

simultaneously with respect to a segmenting contour C and with respect to a piecewise homogeneous approximation u of the image intensities (or the image motion).

Diffusion Snakes

In Chapter 2, we propose to use the external energy E_{ext} of the Mumford-Shah functional [136] and its cartoon limit, which aim at approximating the input image f by a piecewise smooth (or piecewise constant) function u. For the internal energy E_{int} we use the length measure which is typical for curves known as *elastica*, such as the classical snakes. Due to the underlying diffusion process in the energy minimization we named these hybrid models *diffusion snakes*.

We experimentally verify several properties of the diffusion snakes:

- The two problems of image smoothing and optimal contour placement are separated by the two variables u and C in the Mumford-Shah functional. This permits a smoothing of the input image which does not destroy valuable information such as the precise location of edges and corners. The resulting segmentation process can therefore reconstruct the silhouette of an object in a noisy input image without blurring its edges or corners.

- Due to the region-based formulation of the external energy, the contour converges over fairly large distances during the minimization, although there are no balloon-type forces in the functional which would produce

an artificial expansion or contraction of the contour. As shown in example segmentation processes, the contour can therefore both expand and contract for the same parameter choice.

- The internal energy of the elastica generates a rubber-band like behavior of the contour. This prevents the formation of cusps during the evolution of the spline control points.

- The implementation of the segmenting contour as an explicit spline curve and the local optimization by gradient descent permit a relatively fast numerical implementation. Therefore the proposed method is amenable to real-time implementations.

Linear Shape Statistics for Diffusion Snakes

In Chapter 3, we extend the diffusion snake functionals by an internal energy which incorporates a statistical prior on the shape of the segmenting contour. This prior is based on the assumption that the control point vectors associated with a set of training silhouettes are distributed according to a Gaussian probability density.

We present a method of automatically extracting and aligning a set of training shapes, where alignment is done simultaneously with respect to similarity transformations and with respect to renumbering of the control points. We propose to regularize the covariance matrix in order to obtain a Gaussian probability distribution which is nonvanishing in the full space of spline contours. We discuss advantages and disadvantages of this approach over subspace methods such as principal component analysis. In particular, we suggest a choice of the regularizing constant which differs from that proposed in probabilisitic principal component analysis [131, 176]. We argue that due to this regularization, sensible shape priors can be constructed even from very small training sets.

We discuss various methods to incorporate invariance with respect to certain transformations of the shape into the prior. In particular, we present a closed-form variational integration of similarity invariance on the basis of the control point polygons. By aligning the evolving contour to the training set before applying the shape energy, we obtain a variational approach which is entirely parameter-free. In contrast, the local optimization of explicit pose parameters requires additional tuning of associated gradient descent parameters, and introduces additional local minima and possible numerical instabilities. In segmentation tasks we confirm the improved convergence obtained by the closed-form integration of invariance.

Numerical experiments demonstrate several properties of the resulting segmentation process:

- Increasing the weight of the shape prior results in a progressive suppression of unfamiliar shape deformations during the segmentation process.

- Due to the shape prior, the evolving contour is effectively restricted to a submanifold of familiar shapes.

- The successive application of shape priors constructed from different training sets permits to parse a given input image into its constituent components.
- The statistical shape prior compensates for various cases of missing or misleading low-level information, thereby enabling the segmentation of
  - objects in front of a cluttered background,
  - partially occluded objects and
  - objects in images which are corrupted by noise.
- Due to the variational integration of similarity invariance, the evolving contour is entirely free to translate, rotate and expand or shrink, while its shape is always restricted to the domain of familiar shapes.

Nonlinear Shape Statistics Based on Mercer Kernels

In Chapter 4, we introduce a novel statistical shape prior which is based on the assumption that the training shapes are distributed according to a Gaussian density after a nonlinear mapping to an appropriate feature space. Due to the strong nonlinearity, this differs considerably from the former assumption of a Gaussian distribution in the original space. The nonlinearity is modeled implicitely in terms of Mercer kernels [130, 49]. The resulting model constitutes an extension of kernel principal component analysis [164] to a probabilistic framework. It was first proposed in [50] and has more recently also been suggested in [175]. The corresponding nonlinear shape energy contains a single scale parameter, namely the width of the Gaussian kernel σ, which determines the granularity of the model. An automatic estimate of this parameter is presented.

Expressed in the original space, the shape dissimilarity measure can encode arbitrary sets of training shapes. These may form several clusters and banana- or ring-shaped distributions. Numerical experiments demonstrate several properties of the resulting segmentation process:

- Like the linear prior, the nonlinear one suppresses unfamiliar deformations of the evolving contour during the segmentation process.
- In contrast to the linear prior, the nonlinear one can encode in an entirely unsupervised manner shapes of different classes, such as those corresponding to different objects or to different views of a 3D object. The segmentation process permits a precise reconstruction of occluded or cluttered versions of very different simultaneously encoded shapes.
- On the one hand, the prior does not mix shapes of different classes, but on the other hand, it is still a *statistical prior* in the sense that it can generalize to novel views which were not contained in the training set. This is demonstrated by appropriate 2D projections of the training shapes, the evolving contour and the estimated density. The balancing between not mixing different classes and generalizing to novel views is determined by the kernel width σ.

- The nonlinear shape prior is capable of encoding the 3D structure of an object in terms of the silhouettes associated with several 2D views of the object. Incorporated into the segmentation process, it permits to reconstruct in detail the silhouette of a partially occluded 3D figure, viewed from various angles in a tracking experiment with cluttered background.

Motion Competition

Having evaluated different statistical shape dissimilarity models for the internal energy E_{int}, we present a modification of the external energy E_{ext} in Chapter 5. Rather than measuring the piecewise homogeneity of the grey value in a set of regions — as done by the Mumford-Shah functional — we propose to measure the *homogeneity of motion* in the respective regions. Compared to most other methods of motion segmentation, both the evolution of the motion boundary and the estimation of the motion vectors for each region are derived by minimizing a *single* energy functional. The obtained contour evolution equation shows that neighboring regions compete in terms of the respective motion energy densities. We therefore named the resulting segmentation process *motion competition*.

The motion information is determined on the basis of the spatio-temporal derivatives which are obtained from two consecutive images of an image sequence, with no prior smoothing of the input images being neccessary. Motion homogeneity in the separate regions can be defined in terms of various parametric motion models. In our application we detail this for the segmentation of the image plane into regions of *piecewise constant* or *piecewise affine* motion. As in the case of grey value segmentation, we implement the boundary as a closed spline curve. This permits to easily introduce a statistical prior on the shape of the segmenting motion boundary.

Experimental results show several properties of the proposed method:

- The motion segmentation process differs fundamentally from the corresponding grey value segmentation process. Numerous examples show that objects may be segmented on the basis of their *relative motion*, although they are not easily (or even not at all) distinguishable from the background by their *appearance*. In examples of rotating objects, we show that the model of *piecewise constant motion* permits to segment those parts which correspond to the hypothesis of constant motion, whereas the model of *piecewise affine motion* permits a segmentation of the complete object. In contrast, the model of *piecewise constant grey value* may completely fail, for example due to difficult lighting conditions or similar grey values of object and background.

- Minimization of a single energy functional results in the combined optimization of the motion boundary and the motion parameters of each region. The motion estimates are gradually improved, while the contour separates the two different motion fields.

- The method permits to segment differently moving regions, as in the case of a moving object captured by a moving camera. Real-world applications

show that the method is fairly robust to deviations from the parametric model hypothesis, to weak grey value structure and to fairly large motion vectors of several pixels.

- Due to the region-based formulation, the contour converges over fairly large distances to the final segmentation.
- As in the case of grey value segmentation, the introduction of a statistical shape prior effectively restricts the evolving motion boundary to a subspace of familiar shapes. The prior compensates for missing motion information, as due to the aperture problem. In particular, we show that the shape prior enables the segmentation of a moving object even if part of the motion information is occluded by a static object.

6.2 Limitations and Future Work

Improvements on the Shape Metric

A central property of statistical priors is their capacity to generalize and abstract from a finite (possibly very small) set of training samples. In our model, this generalization arises by an interpolation (or morphing) between different training shapes. How "sensible" these interpolated shapes are strongly depends on the metric of the underlying vector space. In our model, the metric is rather simple: It is given by the Euclidean distance of the control point polygons associated with two contours, after alignment with respect to similarity transformation and renumbering of control points. This metric permits fast implementations and works fairly well in most of the applications we have tried.

However, the alignment process will fail to associate corresponding parts as soon as the contour of one of two given objects is stretched or shrunk in one particular area. For example, if a given training shape of a hand is missing one finger, then the equidistant placement of a fixed number of control points during shape acquisition will result in a higher control point density along the remaining fingers, such that a minimization of the control point distance will not permit an alignment of corresponding points. This and other limitations of the simple shape metric are well known from the literature. Numerous results on refined and sometimes highly elaborate shape distances have been proposed — see [9, 196, 78, 116] and the discussion in Section 1.4.

As soon as these distances can be embedded in a Hilbert space structure, the statistical shape models discussed in Chapters 3 and 4 can be applied. One would expect a better generalization capacity, such that the appearance of a given object could be learnt with fewer training samples. In fact, recently Rhodri et al. [60] proposed to jointly solve the problems of shape alignment and statistical learning. There the correct correspondence of contour points is determined by maximizing the "compactness" of the resulting statistical shape model in a minimum description length approach.

Unfortunately most of the more elaborate shape metrics require a costly optimization e.g. by dynamic programming. Since the shape distance and the cor-

responding alignment are a central part not only of the training phase but also of the knowledge-based segmentation, the entire segmentation process would be drastically slowed down. In the future, we will therefore evaluate how a more elaborate shape alignment and shape metric can be efficiently integrated in a statistical shape prior for segmentation. In particular, we will need to evaluate whether the closed-form variational integration of similarity invariance introduced in Section 3.4.2 can be extended to more elaborate distance measures.

Implicit Contour Representations

Closely related to the issue of shape alignment and shape metrics is the question of whether the contour should be represented explicitly or implicitly. As discussed in Section 1.3, both representations have their advantages and drawbacks. Some work on statistical shape models for segmentation with implicit contours has been done (cf. [120, 36, 154]).

However, although implicit contours permit to segment *several* objects in a given image, to our knowledge this has not been done with a statistical shape prior. Moreover, the issues of shape learning and shape alignment are more straight-forward if the contour is given explicitly. We will therefore evaluate how far the methods of alignment, similarity invariance and the variational integration of statistical shape priors can be adapted to implicit contour representations.

Extending the Variational Framework

The goal of the present work was to model the interaction of external visual input and an internally represented, previously acquired statistical model of shape. A segmentation process which takes into account both the external input and the internal knowledge about "permissible" shape variations was obtained by minimizing a joint functional of the form (6.1). As pointed out in the introduction, this variational approach is equivalent to the Bayesian framework of maximum a posteriori estimation.

In Chapters 3, 4 and 5, we showed that minimization of the total energy (6.1) produces a compromise between the segmentation induced by the image information and the one favored by the shape prior. There is no decision process incorporated: The question of whether there really is a hand (or a rabbit or letter) in a given image is never answered.

The variational integration of shape prior and image information facilitates the mathematical modeling, as the segmentation problem is reduced to minimizing functionals which tend to have few local minima. In various examples, we demonstrated that the statistical prior can strongly improve segmentation results. Yet, the proposed fusion of prior knowledge and external information appears different from the way humans tend to incorporate prior knowledge. The latter seems to be far more based on decisions, such as dynamically making and rejecting hypotheses about which object is or is not present in a given image — as can be verified on the Dalmatian in Figure 1.2. Moreover, if some part of the object of interest is occluded, then the occluding section should

not impose a bias on the segmenting contour by drawing it to one side or the other. Essentially one would prefer a segmentation process in which the region homogeneity constraint or the prior shape information can be locally switched off or decreased for certain parts of the segmenting contour. Ideally, such a process should be derived from a probabilistic model which captures the *inter-dependence* of the internal shape prior and the external image information.

Similarly, one should model the fusion of several low-level segmentation cues: As pointed out in the introduction, human observers tend to *choose* a sensible low-level cue for segmentation — e.g. the intensity information for the human silhouette, the texture information for the zebra, or the motion information for the car (cf. Figure 1.1). In Chapter 5, we introduced the cue of motion homogeneity into the Mumford-Shah functional. We compared segmentation results for the models of piecewise constant intensity, piecewise constant motion and piecewise affine motion. However, a central question remains unanswered: How should a machine decide *which* low-level cue is applicable? One way to avoid this decision is to retain all the distinct segmentations obtained by multiple cues, as suggested by Tu and Zhu in [178]. Yet a human observer tends to focus on a segmentation obtained by only *one* of the possible cues, rejecting the other ones. For example, humans tend to inspect the grey value segmentation of the moving rabbit in Figure 5.3, bottom right, for a while, before "understanding" why this corresponds to a consistent segmentation of the given image. In future work, we will therefore focus on more elaborate probabilistic models for the fusion of multiple low-level cues and the statistical shape information.

Region-based Statistical Priors

The Mumford-Shah based segmentation process results from minimizing a functional with respect to both the segmenting contour C and an approximation u of the input image. In this work, we introduced statistical priors of different complexity on the segmenting contour C into the variational approach.

A straight-forward generalization of this idea is to also introduce a statistical prior on the second variable in the functional, namely the region information u. One would expect this to drastically improve the resulting segmentation process, since the contour itself only contains very shallow information about an object of interest. Although priors on the *contour* can strongly improve the segmentation of hands, the silhouettes of a rabbit etc., they would be of little help for segmenting objects such as faces, which are not primarily defined in terms of their outline.

Appearance-based methods for the task of object recognition have been extensively studied (cf. [94]). Statistical models for grey level *appearance*, with a focus on faces, have been proposed among others by Kirby and Sirovich [109]. Combinations of contour and region information have been proposed by Cootes and Taylor [43]. However, the latter approach is based on a parameter fitting process which fits a model to certain image features. In future work, we will instead focus on introducing statistical region or appearance information into region-based variational segmentation methods based on the diffusion snakes.

Appendix A

Effects of the Spline Distance Approximation

Both the alignment process of Section 3.1.4, and the linear statistical model introduced in Section 3.3 are based on the Euclidean distance (3.8), i.e. on the identification of a given contour with its control point polygon. In Section 3.1.2 we argued that the spline distance (3.4) is a more precise measure of the distance between two shapes. Although the effect of working with this more exact distance measure is negligible in practice, we will show for completeness how alignment and linear statistics can be derived on the basis of the spline distance (3.4) and the associated scalar product (3.7).

For the alignment of a set of training shapes we will again employ the simpler complex notation of Section 3.1.4. The distance (3.9) to be minimized is replaced by

$$D^2(z, \hat{z}) = (z - \alpha\hat{z} + \beta)^* \, B \, (z - \alpha\hat{z} + \beta),$$

where B is the matrix (3.6) of spline basis overlaps. Setting the corresponding derivatives to zero, one can solve for the minimizing parameters to obtain:

$$\beta = 0, \quad \alpha = \frac{\hat{z}^* \, B \, z}{\hat{z}^* \, B \, \hat{z}}.$$

As proposed in [11], PCA can also be performed on the basis of this spline distance and the associated scalar product (3.7). The sole modification is that the sample covariance matrix has to be multiplied with the metric matrix A defined in (3.5), otherwise one can proceed as before.

This is most easily seen by transforming the input data to a Euclidean space according to

$$z' = A^{1/2} \, z,$$

Then the sample mean $\bar{z}$ and the sample covariance matrix Σ transform to:

$$\bar{z'} = A^{1/2} \, \bar{z} \qquad \text{and} \qquad \Sigma' = A^{1/2} \, \Sigma \, A^{1/2}.$$

We can perform a sampling similar to (3.14) along the eigenmodes (principal components) associated with the transformed covariance matrix. Transformed back to the control point representation we obtain:

$$z(\alpha_1, \ldots, \alpha_r) = A^{-1/2} \left[\bar{z'} + \sum_{i=1}^{r} \alpha_i \sqrt{\lambda'_i} \, e'_i \right], \tag{A.1}$$

where $\{\lambda'_i, e'_i\}$ denotes the eigensystem of Σ':

$$\Sigma' \, e'_i \;=\; \lambda'_i \, e'_i.$$

With $e_i := A^{-1/2} \, e'_i$ this is equivalent to

$$\Sigma A \, e_i \;=\; \lambda'_i \, e_i, \tag{A.2}$$

and the sampling (A.1) is given by

$$z(\alpha_1, \ldots, \alpha_r) = \bar{z} \;+\; \sum_{i=1}^{r} \alpha_i \, \sqrt{\lambda'_i} \, e_i. \tag{A.3}$$

This expression is the same as the one obtained in (3.14), except that now according to equation (A.2), $\{\lambda'_i, e_i\}$ is the eigensystem of the matrix (ΣA).

However, in practice (for 100 control points), the modification induced by the spline distance have a negligible effect on the eigenmodes. Sampling along the eigenmodes of the matrix (ΣA) produces a shape variation which cannot be distinguished from the one obtained with the simpler Euclidean distance shown in Figure 3.4.

Moreover, for the case of the full Gaussian model, working with the spline distance (3.4) does not modify the shape energy given in (3.17). Expressed in the transformed coordinates the energy reads:

$$\begin{aligned} E'(z') \;&=\; \frac{1}{2} \, (z' - \bar{z'})^t \, \Sigma'^{-1}_{\perp} \, (z' - \bar{z'}) \\ &=\; \frac{1}{2} \, (z - \bar{z})^t A^{1/2} \, A^{-1/2} \Sigma_{\perp}^{-1} A^{-1/2} \, A^{1/2} (z - \bar{z}) \\ &=\; \frac{1}{2} (z - \bar{z})^t \, \Sigma_{\perp}^{-1} \, (z - \bar{z}) \;=\; E(z). \end{aligned}$$

More generally, any probabilistic model for the linearly transformed sample points z' always implies a corresponding model for the control point vectors z. This connection justifies the identification of each shape with its control point vector z, both for the linear statistics in Section 3 and the nonlinear shape statistics introduced in Section 4.

Appendix B

A Multigrid Scheme for Diffusion Snakes

In the case of the full Mumford-Shah model we need to determine the diffusion process which is underlying the contour evolution. The present section gives details on the implementation of a multigrid scheme for solving the corresponding steady state equation

$$\frac{1}{\lambda^2}\frac{dE}{du} \;=\; \frac{1}{\lambda^2}(u-f) \;-\; \nabla\cdot(w_c(x)\,\nabla u) = 0. \tag{B.1}$$

Discretizing this equation by finite differences, we obtain a linear system with Neumann boundary conditions:

$$\boldsymbol{A}u = f, \quad \text{and} \;\; \partial_n u = 0 \text{ on } \partial\Omega.$$

The contour is represented by edgels "between pixels" (micro-edges), such that all image pixels are affected by the diffusion process.

Solving this equation with standard solvers like Gauss-Seidel or Jacobi takes a long time, as low frequencies in the error vanish slowly. Therefore we propose a multigrid implementation, which consists in recursively transfering the problem from a grid with size h to a coarser grid of size $2h$, and solving the corresponding problem on the coarser grid to obtain an improved initialization for the solution on the fine grid.

Standard implementations of numerical multigrid schemes like the one in [171], may easily lead to a poor implementation of the steady state diffusion equation (B.1) due to the strongly inhomogeneous term w_c. The hierarchical representation of this term at multiple scales is even more difficult. For the diffusion snake to work, smoothing across the curve C must be prevented at all scales.

Let v be an approximation to the solution u. Denote the error by $e = u - v$, and the residual by $r = f - \boldsymbol{A}v$. With these notations we obtain for every grid h the following equivalence:

$$\boldsymbol{A}^h u^h = f^h \quad\Longleftrightarrow\quad \boldsymbol{A}^h e^h = r^h.$$

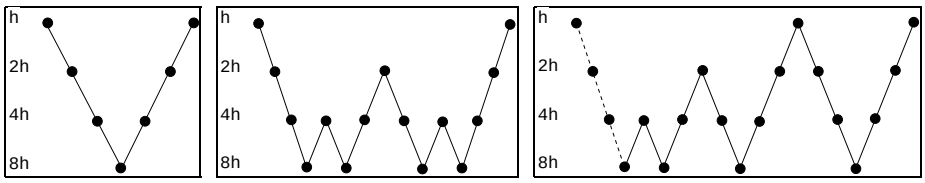


Figure B.1: Schematic diagrams of multigrid cycles. An elegant recursive definition of different multigrid cycles can be found in [26].

In order to solve the latter problem on the fine grid h, we transfer the residual r^h and the matrix $\boldsymbol{A}^h$ to the coarser grid $2h$, solve

$$\boldsymbol{A}^{2h} e^{2h} = r^{2h}$$

for e^{2h}, interpolate back to the fine grid and add e^h to the fine grid solution v^h. This idea is recursively extended to more than two grids, which leads to different multigrid cycles, some of which are depicted in Figure B.1. We found that w-cycles showed the best performance in our experiments.

Interpolation, Restriction and Coarse Grid Representation of $\boldsymbol{A}^h$

Starting with the matrix $\boldsymbol{A}^h$, we need to construct appropriate prolongation operators $\boldsymbol{P}$ and restriction operators $\boldsymbol{R}$, which define the transition from the coarse to the fine grids and vice versa. For this purpose we introduce the stencil notation, where the stencils shown in Table B.1 represent the action of the operator $\boldsymbol{A}$ on a pixel and its 3×3–neighborhood. This notation allows to intuitively understand the effect of the operator $\boldsymbol{A}$ at a given location. The effect of the boundary conditions imposed by the contour and the image edges is given by the zeros in the stencils in Table B.1.

The implementation of the contour as a diffusion border prohibits any restriction or prolongation accross this border. We therefore use matrix-dependent prolongation and restriction operators, as described in [191]. Similar approaches were proposed in [3, 65, 202].

In the following, we will define the prolongation operator, which performs the transition from the coarse grid $2h$ to the fine grid h. According to [191], two constraints have to be fullfilled for the prolongation operator in the one-dimensional case:

$$u^h_{2i} = [\boldsymbol{P}]^{2h}_{i,0} \cdot u^{2h}_i = u^{2h}_i, \tag{B.2}$$

$$(\boldsymbol{A}\boldsymbol{P}u^{2h})_{2i+1} = 0, \tag{B.3}$$

where the first lower index at the stencil denotes the pixel number and the second lower index denotes the position within the stencil, which can be -1, 0 or 1 for left, middle and right in the 1D case.

The first constraint ensures, that all coarse grid points are transfered directly to the finer grid, the second one ensures, that the prolongation operator is

$$[\boldsymbol{A}]_{0,0}=\begin{bmatrix} & 0 & \\ 0 & 1+\frac{2\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix} \quad [\boldsymbol{A}]_{i,0}=\begin{bmatrix} & 0 & \\ -\frac{\lambda^2}{h^2} & 1+\frac{3\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix} \quad [\boldsymbol{A}]_{M,0}=\begin{bmatrix} & 0 & \\ -\frac{\lambda^2}{h^2} & 1+\frac{2\lambda^2}{h^2} & 0 \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix}$$

$$[\boldsymbol{A}]_{0,j}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ 0 & 1+\frac{3\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix} \quad [\boldsymbol{A}]_{i,j}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ -\frac{\lambda^2}{h^2} & 1+\frac{4\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix} \quad [\boldsymbol{A}]_{M,j}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ -\frac{\lambda^2}{h^2} & 1+\frac{3\lambda^2}{h^2} & 0 \\ & -\frac{\lambda^2}{h^2} & \end{bmatrix}$$

$$[\boldsymbol{A}]_{0,N}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ 0 & 1+\frac{2\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & 0 & \end{bmatrix} \quad [\boldsymbol{A}]_{i,N}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ -\frac{\lambda^2}{h^2} & 1+\frac{3\lambda^2}{h^2} & -\frac{\lambda^2}{h^2} \\ & 0 & \end{bmatrix} \quad [\boldsymbol{A}]_{M,N}=\begin{bmatrix} & -\frac{\lambda^2}{h^2} & \\ -\frac{\lambda^2}{h^2} & 1+\frac{2\lambda^2}{h^2} & 0 \\ & 0 & \end{bmatrix}$$

Table B.1: Stencils for diffusion snakes. Each stencil defines the action of operator $\boldsymbol{A}$ at a given pixel location, where the zeros denote the effect of a diffusion boundary.

adapted to the matrix $\boldsymbol{A}$. The odd pixels of the finer grid can then be obtained from equation (B.3):

$$u^h_{2i+1} = -\frac{[\boldsymbol{A}]^h_{(2i+1),-1} \cdot u^{2h}_i + [\boldsymbol{A}]^h_{(2i+1),1} \cdot u^{2h}_{i+1}}{[\boldsymbol{A}]^h_{(2i+1),0}}.$$

A similar solution for the prolongation $\boldsymbol{P}$ is obtained in the two-dimensional case [191]. The stencils for the restriction correspond to the prolongation stencils, normalized so that they sum up to 1. Further details can be found in [177].

With these definitions of prolongation $\boldsymbol{P}$ and restriction $\boldsymbol{R}$ from the matrix $\boldsymbol{A}^h$, we construct the coarse grid matrix $\boldsymbol{A}^{2h}$ by using Galerkin coarsening:

$$\boldsymbol{A}^{2h} = \boldsymbol{R}\boldsymbol{A}^h\boldsymbol{P}.$$

To avoid a full matrix multiplication, we exploit the stencil notation as done in the efficient algorithm CALRAP [191]. Given $\boldsymbol{A}^{2h}$, we can then construct prolongation and restriction for the next coarser level and so on.

Appendix C

From Feature Space Distances to Classical Methods of Density Estimation

In recent years the feature spaces induced by Mercer kernels have become a popular framework for classification and regression estimation, giving rise to methods such as Support Vector Machines and kernel PCA. In Section 4.4, we derived an extension of kernel PCA to a probabilistic framework by modeling the distribution of a set of sample vectors upon a nonlinear mapping ϕ from the original space $\mathbb{R}^n$ to a feature space Y. We assumed that the data upon mapping to the generally higher-dimensional space Y are distributed according to a Gaussian probability density. We then used the associated energy (given by the negative logarithm of the probability) as a dissimilarity measure. It is a Mahalanobis type distance in the feature space Y:

$$E_\phi(z) \;=\; \big(\phi(z) - \phi_0\big)^t \;\; \Sigma_\phi^{-1} \; \big(\phi(z) - \phi_0\big). \tag{C.1}$$

We implicitely chose a particular family of nonlinear mappings ϕ by specifying the scalar product of two mapped points in the space Y in terms of the Mercer kernel

$$k(x,y) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \quad x, y \in \mathbb{R}^n. \tag{C.2}$$

Due to the strong nonlinearity, the obtained energy expressed in the original space is no longer quadratic. In fact, numerical experiments showed that the associated level lines of constant energy can have essentially arbitrary form. Figure 4.6 demonstrates that one can model distributions consisting of several clusters which by themselves can be ring- or banana-shaped.

Although by no means exhaustive, the following sections will specify some relations between distances in feature space and classical methods of density estimation. We will show that the Euclidean distance in feature space is equivalent to a Parzen estimator in the original space (up to normalization). We then analyze the Mahalanobis distance (C.1) in the original space and point out similarities to traditional methods in the field of density estimation.

C.1 Relation to the Parzen Estimator

Density estimation deals with the approximation of an unknown density from a set of iid random sample vectors. Generally one distinguishes parametric and nonparametric density estimation. This terminology is somewhat misleading, since in practice one determines certain parameters for both approaches. Following [165], we shall therefore call an estimator *nonparametric*, iff the influence of any given sample vector on the estimated density is relevant on a *local* scale only. Conversely, for a *parametric* estimator, a given sample vector may have a *global* influence on the estimated density.

A typical representative of a nonparametric estimator is the Parzen estimator [74, 2, 151, 144, 29]:

$$E_{parzen}(x) = \frac{1}{m\sigma^n} \sum_{i=1}^{m} K\left(\frac{x - x_i}{\sigma}\right), \tag{C.3}$$

where $\chi = \{x_i\}_{i=1,\ldots,m}$ is the set of sample vectors $x_i \in \mathbb{R}^n$, and K is a Borel measurable kernel function, which is nonnegative and integrates to 1. The width σ of the kernel is typically chosen as a function of the sample size m and of the data set χ. The Parzen estimator (C.3) has been extensively studied, results on consistency, stability and the rate of convergence have been obtained. For an overview we refer to [66, 167].

By the following proposition, we present a different interpretation of the Parzen estimator, namely we prove that — apart from normalization — the Parzen estimator is equivalent to the Euclidean distance $E = -\log \mathcal{P}$ associated with a spherical distribution (isotropic Gaussian) $\mathcal{P}$ in an appropriate feature space.

Proposition 1 *Let $\chi = \{x_i\}_{i=1,\ldots,m}$ be a set of sample vectors $x_i \in \mathbb{R}^n$, let $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a translation-invariant (or stationary) Mercer kernel:*

$$k(x, y) = \frac{1}{\sigma^n} K\left(\frac{x - y}{\sigma}\right), \tag{C.4}$$

with a function K which is positive and integrates to 1, and a constant $\sigma > 0$. Let $\phi : \mathbb{R}^n \to Y$ be an associated mapping of the sample vectors $x_i \in \chi$ to a feature space Y. Let $\mathcal{P}$ be the isotropic (spherical) Gaussian probability distribution in Y estimated from the mapped sample vectors.

Then the corresponding energy $E = -\log(\mathcal{P})$ (Euclidean distance in feature space) is equivalent to a Parzen estimator in the original space $\mathbb{R}^n$ (up to scaling and an additive constant).

Proof: The isotropic Gaussian probability estimate in Y for the set of mapped sample vectors is given by

$$\mathcal{P}(\phi) \propto \exp\left(-\frac{|\phi - \phi_0|^2}{2\rho^2}\right),$$

where ϕ_0 denotes the sample mean of the mapped vectors (4.2) and ρ the sample variance. The corresponding energy — up to scaling and a constant — is quadratic in ϕ:

$$\tilde{E}(\phi) = |\phi - \phi_0|^2 .$$

Using the Mercer identity $k(x, y) = (\phi(x), \phi(y))$, this energy can be expressed as a function in the original space:

$$E(x) = \tilde{E}(\phi(x)) = k(x, x) - \frac{2}{m} \sum_{i=1}^{m} k(x, x_i) + \frac{1}{m^2} \sum_{i,j=1}^{m} k(x_i, x_j). \tag{C.5}$$

With (C.4) we obtain the final result:

$$E(x) = \text{const.} - \frac{2}{m\sigma^n} \sum_{i=1}^{m} K\left(\frac{x - x_i}{\sigma}\right). \tag{C.6}$$

Up to scaling and a constant, this is the Parzen estimator (C.3). □

Remarks: Under the above assumptions, the existence of the mapping ϕ in Proposition 1 is guaranteed by the Mercer theorem. The scale and the additive constant shall not be further investigated, because they are irrelevant in many applications of density estimation, such as clustering or minimizing the above energy as a dissimilarity measure. If desired, they can be numerically determined: Evaluation at positions far from the data set gives the additive constant. Upon subtraction of this constant, the scale can be determined by numerical integration.

Note that the correspondence to feature space estimators is *not* produced by a standard probability density transformation which holds for invertible functions ϕ:

$$\mathcal{P}(x) = \mathcal{P}(\phi(x)) \ |\det D\phi(x)| .$$

In our approach the feature space enters by a generally *nonlinear* and *non-invertible* mapping ϕ. And the correspondences are defined directly in terms of the associated energies as proposed in (C.5). Yet, this correspondence to feature space density estimates leads to a new interpretation and a possible generalization of the classical Parzen estimator.

Recently, a similar link between a spherical distribution in feature space and a Parzen-like estimator was established in [13]. There, the authors propose to estimate the *smallest enclosing sphere* in feature space for a given set of mapped data points. The differences are mainly the following: Our approach is a probabilistic one. Therefore the center ϕ_0 of the sphere in our model corresponds to the mean of the mapped data points and consequently we obtain a Parzen estimator in the original space. The smallest enclosing sphere [13, 163] does not correspond to a probabilistic approach such that the respective center ϕ_0 is generally not identical with the mean of the mapped data and consequently the estimator obtained in [13] is generally not the same as the Parzen estimator. In fact, it is a Parzen estimator if and only if the center of the sphere conincides with the mean of the mapped sample vectors.

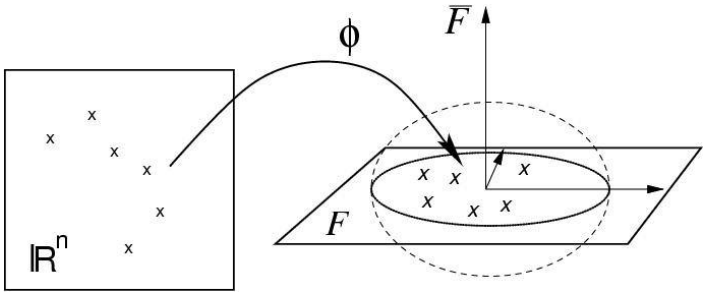


Figure C.1: Schematic diagram of the mapping to the feature space $Y = F \bigoplus \overline{F}$ and the estimated Gaussian, where F is the subspace spanned by the mapped sample vectors.

C.2 Remarks on the Anisotropic Gaussian

Motivated by this connection between a spherical distribution in the feature space Y and the Parzen estimator in the original space, we will investigate the generalization from the spherical distribution in feature space to an ellipsoidal one, shown in the schematic diagramm of Figure C.1. More precisely, we will investigate the energy corresponding to the feature space density

$$\mathcal{P}(\phi) \propto \exp\left(-\frac{1}{2}\,\tilde{\phi}^{\,t}\,\Sigma^{-1}\,\tilde{\phi}\right), \tag{C.7}$$

where $\tilde{\phi} := \phi - \phi_0$ denotes centering in the feature space Y, with ϕ_0 being the mean of a set of mapped sample vectors as defined in (4.2). As discussed in Section 4.4, we will regularize the sample covariance matrix $\tilde{\Sigma}$ in the feature space Y, defined in (4.3), by replacing the zero eigenvalues with a small positive constant $\lambda_\perp$:

$$\Sigma = V\,\Lambda\,V^t \;+\; \lambda_\perp\left(I \;-\; V\,V^t\right) = \tilde{\Sigma} \;+\; \lambda_\perp\left(I \;-\; V\,V^t\right), \tag{C.8}$$

where Λ denotes the diagonal matrix of ordered nonzero eigenvalues $\lambda_1, \ldots, \lambda_r$ of $\tilde{\Sigma}$ and V the matrix of corresponding eigenvectors $V_1, \ldots, V_r$. The regularizing constant is chosen in the range $\lambda_\perp \in (0, \lambda_r)$. For a detailed discussion, we refer to Section 3.3.

Proposition 2 *Let $\chi = \{x_i\}_{i=1,\ldots,m}$ be a set of sample vectors $x_i \in \mathbb{R}^n$, let $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a Mercer kernel with properties as required in Proposition 1. Let $\phi : \mathbb{R}^n \to Y$ be an associated mapping of the sample vectors $x_i \in \chi$ to a feature space Y.*
Let $\mathcal{P}$ be the anisotropic (ellipsoidal) Gaussian probability distribution (C.7) estimated from the mapped sample vectors, with the sample covariance matrix regularized as in (C.8).

Then the corresponding distance (or energy) $E = -\log(\mathcal{P})$, expressed in the original space $\mathbb{R}^n$, is (up to scaling and a constant) given by

$$E(x) = A(x) + B(x),$$

such that:

(i) $A > 0$ *is given by the Parzen-like estimator (C.6),*

(ii) $B(x) < 0 \;\; \forall x \in \mathbb{R}^n$,

(iii) $|B(x)| \leq |A(x)| \;\; \forall x \in \mathbb{R}^n$,

Proof: As in the isotropic case, the energy associated with (C.7) is quadratic in Y. Up to scale and an additive constant, it is given by

$$\tilde{E}(\phi) = \lambda_\perp \tilde{\phi}^t \, \Sigma_\phi^{-1} \, \tilde{\phi},$$

where $\tilde{\phi}$ denotes centering with respect to the mapped sample vectors as introduced in (4.4), and rescaling by $\lambda_\perp$ was done for simplicity. Using definition (C.8), we get

$$\tilde{E}(\phi) = \tilde{\phi}^t \left(\lambda_\perp V \Lambda^{-1} V^t + (I - V V^t) \right) \tilde{\phi}.$$

Expressed in the original space $\mathbb{R}^n$, this reads:

$$E(x) = \tilde{E}\left(\phi(x)\right) = \sum_{k=1}^{r} \frac{\lambda_\perp}{\lambda_k} \left(V_k, \tilde{\phi}(x) \right)^2 \; + \; |\tilde{\phi}(x)|^2 \; - \; \sum_{k=1}^{r} \left(V_k, \tilde{\phi}(x) \right)^2 .$$

Separating isotropic and anisotropic components we obtain:

$$E(x) = |\tilde{\phi}(x)|^2 \; + \; \sum_{k=1}^{r} \left(\frac{\lambda_\perp}{\lambda_k} - 1 \right) \left(V_k, \tilde{\phi}(x) \right)^2 . \tag{C.9}$$

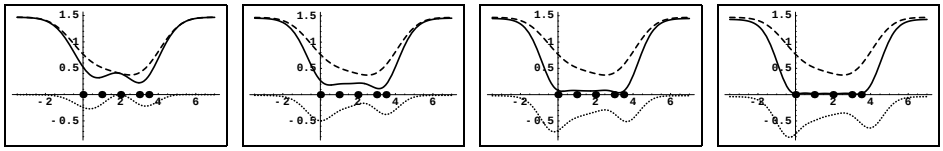
Denote the first term by $A(x)$ and the last one by $B(x)$. We immediately see that $A > 0$ is given by the Parzen-like estimator in (C.6). It corresponds to the largest sphere which can be fit in the regularized ellipsoid shown in the schematic diagramm of Figure C.1.

Since $\lambda_\perp \leq \lambda_k$, $k = 1, \ldots, r$, we have $B < 0$. Moreover, the absolute value of B is bounded by A:

$$|B(x)| \; \leq \; \sum_{k=1}^{r} \left| 1 - \frac{\lambda_\perp}{\lambda_k} \right| \left(V_k, \tilde{\phi}(x) \right)^2 \; \leq \; \sum_{k=1}^{r} \left(V_k, \tilde{\phi}(x) \right)^2 \; \leq \; |\tilde{\phi}(x)|^2 \; = \; A(x)$$

This concludes the proof. □

This connection to the Parzen estimator justifies the use of stationary kernel functions k such as the Gaussian kernel (C.2), rather than any of the other kernels presented in Section 4.3.3. Moreover, it also justifies the estimation of the kernel width σ by similar methods as used for the Parzen estimator — see Section 4.4.4. In the case of the Parzen estimator, more elaborate estimates have been proposed, based on asymptotic expansions such as the parametric method [62], heuristic estimates [181, 166], or maximum likelihood optimization by cross validation [68, 37]. See [66] for an overview.



$r=1$ eigenmode $r=2$ eigenmodes $r=3$ eigenmodes $r=4$ eigenmodes

Figure C.2: Effect of various feature space eigenmodes on the estimated density. Five data points are indicated by big dots (•). The **dashed line** shows the isotropic component (Parzen-like estimate). The **dotted line** shows the anisotropic component given by the second part of equation (C.9) for increasing number r of eigenmodes. The **solid line** gives the sum of isotropic and anisotropic components.

C.3 Numerical Analysis of the Anisotropic Gaussian

In Section C.2 we showed that due to the regularization (C.8) of the covariance matrix the energy corresponding to a Gaussian density in feature space splits into an isotropic and an anisotropic component given by the two terms in equation (C.9). We showed that, expressed in the original space, the isotropic part corresponds to a Parzen-like estimate.

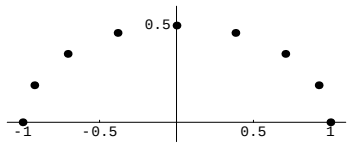
The anisotropic component, i.e. the second term in (C.9), energetically favors positions x which — via the nonlinear mapping ϕ — correspond to large eigenmodes in the feature space Y.

To visualize the effect of these eigenmodes on the estimated density, Figure C.2 shows the isotropic component and the contribution of the first r feature space eigenmodes to the estimated density for $r = 1, \ldots, 4$ for a set of five 1D data points — see (C.9). The kernel width was fixed to $\sigma = \mu$ with μ as defined in (4.21).

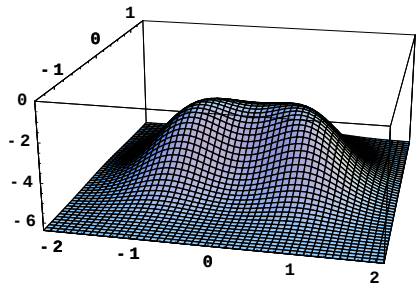
Qualitatively, the effect of the anisotropic component in the given example can be described as follows: The Parzen estimator (isotropic component shown by the dashed line in Figure C.2, right side) tends to favor areas with more data points such as the central part of the data distribution and the right side where two data points are closer. The anisotropic component (dotted line in Figure C.2, right side) compensates this effect in such a way that the estimated density is fairly constant in the area of data points (solid line in Figure C.2, right side).

A similar effect can be observed in two-dimensional examples. Figure C.3 shows 9 data points which are equidistantly distributed on a semiellipse, the isotropic and the anisotropic components of the density estimate and the sum of both. Note that the two components are balanced in such a way that the arc-like structure is well captured by the total estimate (bottom right), even with a very small number of sample points. The kernel width was fixed to $\sigma = 1.5\mu$ for the $2D$-examples, with μ as defined in (4.21).

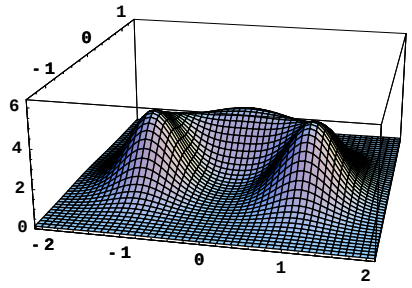
Figure C.4 shows a small number of data points arranged along the figure of a cross. Again the isotropic and anisotropic components are balanced in such a



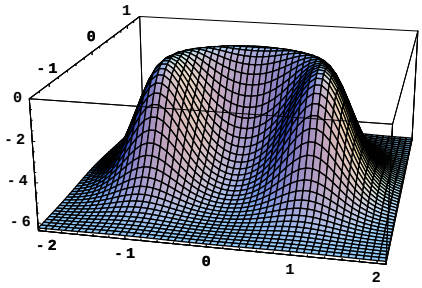
Sample vectors on an arc



Isotropic component (Parzen)



Anisotropic component



Sum of both components
(Generalized Parzen)

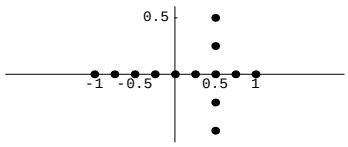
Figure C.3: Effect of anisotropic component for 2D data. Nine data points distributed on an arc (top left), isotropic component of density estimate (top right), anisotropic component (bottom left), and total estimate (bottom right). Note that the structure of the elliptical arc is well captured even though the number of data points is quite small. For better visibility, all energies were inverted.

way that the cross-like structure is well captured by the total estimate (bottom right). Note that, compared to the Parzen estimate, this generalized Parzen estimate produces ridges of fairly constant height, even though the number of sample points is small.

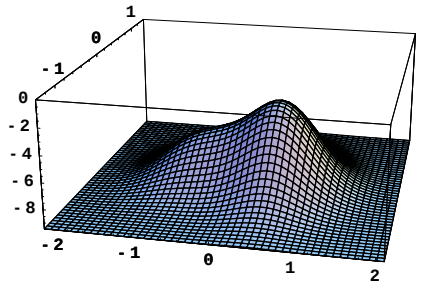
Figure C.5 shows the example of a set of 2D points which were randomly sampled along two spirals (left). Middle and right image show the Parzen and the generalized Parzen for appropriate values of the kernel width σ. Note that the spiral structures are more pronounced by the generalized Parzen.

C.4 Relation to Other Approaches

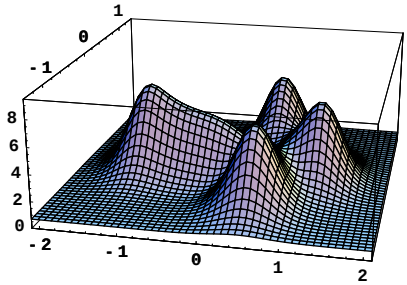
A number of generalizations of the original Parzen estimator (C.3) have been proposed. Although the generalization proposed in this paper is derived from completely different considerations — namely from correspondences to simple parametric density estimates in appropriate feature spaces — we will show that in the case of the Gaussian kernel (C.2), there are certain similarities of the final expression to other extensions of the Parzen estimator.



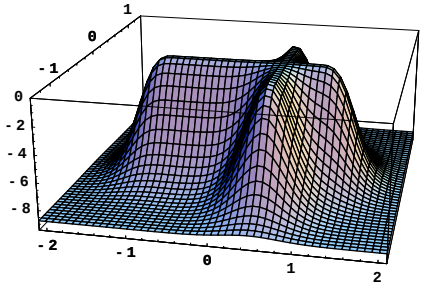
Set of sample vectors



Isotropic component (Parzen)

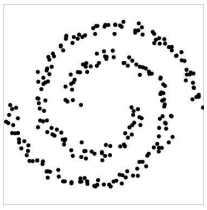


Anisotropic component

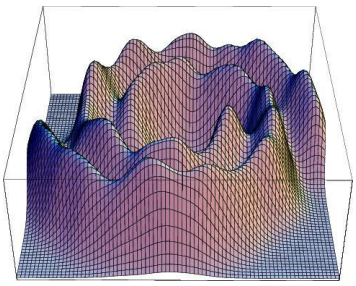


Sum of both components
(Generalized Parzen)

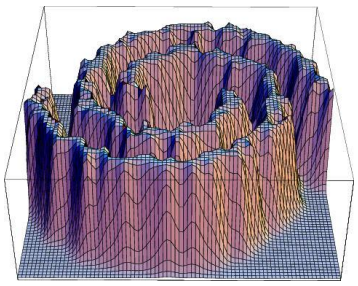
Figure C.4: Effect of anisotropic component for 2D data. Data points distributed on a cross (top left). The Parzen estimator (top right) tends to favor the intersection of the two structures. The anisotropic component (bottom left) compensates this in such a way that the total estimate (bottom right) is fairly constant along the cross-like structure. For better visibility, all energies were inverted.



Sample vectors



Parzen



Generalized Parzen

Figure C.5: Sample vectors randomly distributed on two spirals **(left)**, corresponding estimates of Parzen **(middle)** and generalized Parzen **(right)** for appropriate values of the kernel width σ.

To this end, we will express the energy (C.9) in terms of the kernel function (C.4). Using the expansion (4.10) of the feature space eigenvectors V_k and the definition of $\tilde{\phi}$ in (4.4), we get:

$$\begin{aligned} E(x) &= \left(\tilde{\phi}(x)\right)^2 + \sum_{k=1}^{r} \left(\frac{\lambda_\perp}{\lambda_k} - 1\right) \left(V_k, \tilde{\phi}(x)\right)^2 \\ &= \left(\tilde{\phi}(x)\right)^2 + \sum_{k=1}^{r} \left(\frac{\lambda_\perp}{\lambda_k} - 1\right) \left(\sum_{i=1}^{m} \alpha_i^k \left(\tilde{\phi}(x_i), \tilde{\phi}(x)\right)\right)^2 \\ &= (\phi(x)-\phi_0)^2 + \sum_{k=1}^{r} \left(\frac{\lambda_\perp}{\lambda_k} - 1\right) \left(\sum_{i=1}^{m} \alpha_i^k \big((\phi(x_i)-\phi_0), (\phi(x)-\phi_0)\big)\right)^2. \end{aligned}$$

Using the definition of ϕ_0 in (4.2) and the Mercer identity (4.1), we obtain:

$$\begin{aligned} E(x) &= k(x,x) - \frac{2}{m}\sum_{i=1}^{m} k(x,x_i) + \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i,x_j) \\ &+ \sum_{k=1}^{r}\left(\frac{\lambda_\perp}{\lambda_k}-1\right)\left[\sum_{i=1}^{m}\alpha_i^k\left(k(x,x_i) - \frac{1}{m}\sum_{s=1}^{m}(k(x_i,x_s)+k(x,x_s)) + \frac{1}{m^2}\sum_{s,t=1}^{m} k(x_s,x_t)\right)\right]^2. \end{aligned}$$

This final result is of the form:

$$E(x) = \sum_{i=1}^{m} a_i\, k_\sigma(x,x_i) + \sum_{i,j=1}^{m} b_{ij}\, k_{\tilde{\sigma}}\left(x, \frac{x_i+x_j}{2}\right) + \text{const.}, \tag{C.10}$$

with coefficients $\{a_i\}_{i=1,\ldots,m}$ and $\{b_{ij}\}_{i,j=1,\ldots,m}$ which are determined from the sample vectors. In (C.10), we indicated the width of the kernel by an index. Moreover, we used the fact that for the Gaussian kernel (C.2), the product of two kernels will again give a kernel, which is centered in the middle of these two kernels and of a smaller width $\tilde{\sigma} = \sigma/\sqrt{2}$:

$$\begin{aligned} k_\sigma(x,x_i)\, k_\sigma(x,x_j) &= \frac{1}{\sigma^{2n}} e^{-\frac{1}{2\sigma^2}\left[(x-x_i)^2+(x-x_j)^2\right]} \\ &= k_{\tilde{\sigma}}\left(x, \frac{x_i+x_j}{2}\right) \frac{1}{(2\tilde{\sigma})^n}\, e^{-\frac{1}{2\tilde{\sigma}^2}\left(\frac{x_i-x_j}{2}\right)^2}. \end{aligned}$$

Conceptually, the generalized Parzen estimator (C.10) differs from the original Parzen estimator (C.3) in three ways: First, the kernels in (C.10) do not necessarily have the same weights. Secondly, additional kernels are located at the respective center of each pair of sample points. And thirdly, we have kernels of two different widths σ and $\tilde{\sigma}$.

These three modifications to the original Parzen estimator have been separately proposed. Kernels of variable width were introduced in [24]. Kernels of different weight and center locations differing from the sample vectors appear, for example, as a maximum likelihood estimate for the *Convolution Sieve* proposed in [82, 80]. For the case of 1-D data $\{x_i\}_{i=1,\ldots,m}$, they obtain an estimate of the form

$$E_{conv.sieve}(x) = \sum_{i=1}^{\tilde{m}} p_i \frac{1}{\sigma} K\left(\frac{x-y_i}{\sigma}\right),$$

with weights p_i and kernel centers y_i which are strictly contained in the interval $(\min_i x_i, \max_i x_i)$. The values p_i and y_i are then determined in an optimization procedure. Similarly, in our case of n-dimensional data, the centers of all kernels in estimate (C.10) are strictly contained in the convex hull spanned by the sample vectors.

Note that although these three modifications were previously proposed, they arise quite naturally when extending the spherical density estimate in feature space to an ellipsoidal one.

Bibliography

[1] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[2] H. Akaike. An approximation to the density function. *Ann. Inst. Statist. Math.*, 6:127–132, 1954.

[3] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comp.*, 2(4):430–454, 1981.

[4] L. Alvarez, F. Guichard, P. L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Arch. Rat. Mech. Anal.*, 123:199–257, 1993.

[5] L. Ambrosio. A compactness theorem for a special class of functions of bounded variation. *Boll. Un. Mat. Ital.*, 3-B:857–881, 1989.

[6] L. Ambrosio, N. Fusco, and D. Pallara. Partial regularity of free discontinuity sets,ii. *Ann. Scuola Norm. Sup. Pisa Cl. Sci.*, 24:39–62, 1997.

[7] L. Ambrosio and V. M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ–convergence. *Comm. Pure Appl. Math.*, 43:999–1036, 1990.

[8] K. Arbter, W. E. Snyder, H. Burkhardt, and C. Hirzinger. Application of affine–invariant Fourier descriptors to recognition of 3–d objects. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 12(7):640–647, 1990.

[9] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[10] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In J.O. Eklundh, editor, *Proc. of the Europ. Conf. on Comp. Vis.*, volume 801 of *LNCS*, pages 316–327. Springer-Verlag, 1994.

[11] A. Baumberg and D. Hogg. An adaptive eigenshape model. In *Proc of the 6th Brit. Mach. Vis. Conf.*, volume 1, pages 87–96, 1995.

[12] R. E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1957.

[13] A. Ben-hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *J. of Machine Learning Res.*, 2001. to appear.

[14] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 13(8):775–790, 1991.

[15] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise–smooth flow fields. *Comp. Vis. Graph. Image Proc.: IU*, 63(1):75–104, 1996.

[16] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. J. of Comp. Vis.*, 11(2):127–145, 1993.

[17] A. Blake and M. Isard. *Active Contours*. Springer, London, 1998.

[18] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[19] A. Bonnet. Sur la régularité des bords des minima de la fonctionelle de Mumford–Shah. *C. R. Acad. Sci. Paris*, t.321, Série I:1275–1279, 1995.

[20] A. Bonnet. *Cracktip is a global Mumford-Shah minimizer*. Société Mathématique de France, Paris, 2001.

[21] F. L. Bookstein. *The Measurement of Biological Shape and Shape Change*, volume 24 of *Lect. Notes in Biomath.* Springer, New York, 1978.

[22] F. L. Bookstein. Principal warps: thin–plate splines and the decomposition of deformations. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 11:567–585, 1989.

[23] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of the 5th Annual ACM Workshop on Comput. Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

[24] L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19:135–144, 1977.

[25] C. Brice and C. Fennema. Scene analysis using regions. *Artif. Intell.*, 1:205–226, 1970.

[26] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Siam, Philadelphia, second edition, 2000.

[27] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[28] H. Burkhardt. *Transformationen zur lageinvarianten Merkmalsgewinnung*, volume 10(7) of *Fortschritt–Berichte*. VDI Verlag, Düsseldorf, 1979.

[29] T. Cacoullos. Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 18:896–904, 1966.

[30] J. Canny. A computational approach to edge detection. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 8(6):679–698, November 1986.

[31] V. Caselles, F. Catté, T. Coll, and F. Dibos. A gemoetric model for active contours in image processing. *Numer. Math.*, 66:1–31, 1993.

[32] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. IEEE Internat. Conf. on Comp. Vis.*, pages 694–699, Boston, USA, 1995.

[33] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 19(4):394–398, 1997.

[34] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29(1):182–193, 1992.

[35] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford–Shah functional in image processing. In *IEEE Workshop on Variational and Level Set Methods*, pages 161–168, Vancouver, CA, 2001.

[36] Y. Chen, S. Thiruvenkadam, H. Tagare, F. Huang, D. Wilson, and E. Geiser. On the incorporation of shape priors into geometric active contours. In *IEEE Workshop on Variational and Level Set Methods*, pages 145–152, Vancouver, CA, 2001.

[37] Y. S. Chow, S. Geman, and L. D. Wu. Consistent cross-validated density estimation. *Annals of Statistics*, 11:25–38, 1983.

[38] G. C.-H. Chuang and C.-C. J. Kuo. Wavelet descriptors of planar curves: Theory and applications. *IEEE Trans. on Image Processing*, 5(1):56–70, 1996.

[39] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. IEEE Internat. Conf. on Comp. Vis.*, pages 616–625. Springer, 1990.

[40] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 15(11):1131–1147, 1993.

[41] T. Cootes. Statistical models of appearance for computer vision. techn. report, Wolfson Image Analysis Unit, University of Manchester, UK, 2001.

[42] T. Cootes and C. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. In J. Illingworth, editor, *Brit. Mach. Vis. Conf.*, pages 639–648, 1993.

[43] T. Cootes and C. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, pages 236–248, 2001.

[44] T. Cootes, C. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Brit. Mach. Vis. Conf.*, pages 9–18, 1992.

[45] T. Cootes, C. Taylor, A. Lanitis, D. Cooper, and J. Graham. Building and using flexible models incorporating grey–level infromation. In *Proc. 4th Int. Conf. on Computer Visison*, pages 242–246, 1993.

[46] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–365, 1994.

[47] T. F. Cootes, C. J. Taylor, D. M. Cooper, and J. Graham. Active shape models – their training and application. *Comp. Vision Image Underst.*, 61(1):38–59, 1995.

[48] T.F. Cootes and C.J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–574, 1999.

[49] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, Inc., New York, 1953.

[50] D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics via kernel spaces. In B. Radig and S. Florczyk, editors, *Pattern Recognition*, volume 2191 of *LNCS*, pages 269–276, Munich, Germany, Sept. 2001. Springer.

[51] D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics in Mumford–Shah based segmentation. In A. Heyden et al., editors, *Proc. of the Europ. Conf. on Comp. Vis.*, volume 2351 of *LNCS*, pages 93–108, Copenhagen, May 2002. Springer, Berlin.

[52] D. Cremers, T. Kohlberger, and C. Schnörr. Shape statistics in kernel space for variational image segmentation. *Patt. Recog.*, 2002. To appear.

[53] D. Cremers and C. Schnörr. Motion Competition: Variational integration of motion segmentation and shape regularization. In L. van Gool, editor, *Pattern Recognition*, volume 2449 of *LNCS*, pages 472–480, Zürich, Sept. 2002. Springer.

[54] D. Cremers and C. Schnörr. Statistical shape knowledge in variational motion segmentation. In A. Pece, Y. N. Wu, and R. Larsen, editors, *1st Internat. Workshop on Generative-Model-Based Vision*, Copenhagen, June, 2 2002. Univ. of Copenhagen. http://www.diku.dk/research/published/2002/02-01.

[55] D. Cremers and C. Schnörr. Statistical shape knowledge in variational motion segmentation. *Image and Vision Computing*, 21(1):77–86, 2003.

[56] D. Cremers, C. Schnörr, and J. Weickert. Diffusion snakes: Combining statistical shape knowledge and image information in a variational framework. In *IEEE First Workshop on Variational and Level Set Methods*, pages 137–144, Vancouver, 2001.

[57] D. Cremers, C. Schnörr, J. Weickert, and C. Schellewald. Diffusion snakes using statistical shape knowledge. In G. Sommer and Y.Y. Zeevi, editors, *Algebraic Frames for the Perception-Action Cycle*, volume 1888 of *LNCS*, pages 164–174, Kiel, Germany, Sept. 10–11, 2000. Springer.

[58] D. Cremers, C. Schnörr, J. Weickert, and C. Schellewald. Learning of translation invariant shape knowledge for steering diffusion snakes. In G. Baratoff and H. Neumann, editors, *Dynamische Perzeption*, volume 9 of *Proceedings on Artificial Intelligence*, pages 117–122, Ulm Germany, Nov. 2000. Infix.

[59] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion Snakes: Introducing statistical shape knowledge into the Mumford–Shah functional. *Int. J. of Comp. Vis.*, 50(3):295–313, 2002.

[60] R. Davies, C. Twining, Cootes T., J. Waterton, and C. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Trans. on Medical Imaging*, 2002. To appear.

[61] E. De Giorgi, M. Carriero, and A. Leaci. Existence theorems for a minimum problem with free discontinuity set. *Arch. Rat. Mech. Anal.*, 108:195–218, 1989.

[62] P. Deheuvels. Estimation non paramétrique de la densité par histogrammes généralisés. *Revue de Statistique Appliquée*, 25:5–42, 1977.

[63] H. Delingette. On smoothness measures of active contours and surfaces. In *IEEE Workshop on Variational and Level Set Methods*, pages 43–50, Vancouver, CA, 2001.

[64] H. Delingette and J. Montagnat. New algorithms for controlling active contours shape and topology. In D. Vernon, editor, *Proc. of the Europ. Conf. on Comp. Vis.*, volume 1843 of *LNCS*, pages 381–395. Springer, 2000.

[65] J. E. Dendy. Black box multigrid. *J. Comp. Phys.*, 48:366–386, 1982.

[66] L. Devroye and L. Györfi. *Nonparametric Density Estimation. The L1 View.* John Wiley, New York, 1985.

[67] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* Wiley, Chichester, 1998.

[68] R. P. W. Duin. On the choice of smoothing parameters for Parzen estimators of probability density functions. *IEEE Trans. on Computers*, 25:1175–1179, 1976.

[69] N. Dunford and J. T. Schwartz. *Linear Operators: Part II: Spectral Theory, Self Adjoint Operators in Hilbert Space.*, volume VII of *Pure and Applied Mathematics*. John Wiley & Sons, New York, 1963.

[70] N. Duta, A. Sonka, and Jain. A. K. Learning shape models from examples using automatic shape clustering and procrustes analysis. In A. Kuba, M. Samal, and A. Todd-Pokropek, editors, *Proc. Inf. Proc. in Med. Imaging*, volume 1613 of *LNCS*, pages 370–375. Springer, 1999.

[71] G. Farin. *Curves and Surfaces for Computer–Aided Geometric Design*. Academic Press, San Diego, 1997.

[72] G. Farnebäck. *Spatial Domain Methods for Orientation and Velocity Estimation*. PhD thesis, Dept. of Electrical Engineering, Linköpings universitet, 1999.

[73] G. Farnebäck. Very high accuracy velocity estimation using orientation tensors, parametric motion, and segmentation of the motion field. In *Proc. 8th ICCV*, volume 1, pages 171–177, 2001.

[74] E. Fix and J. L. Hodges. Nonparametric discrimination: Consistency properties. Technical Report 11, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[75] D. Forsyth, J. Mundy, A. Zisserman, and C. Brown. Projectively invariant representations using implicit algebraic curves. In O. Faugeras, editor, *Proc. of the Europ. Conf. on Comp. Vis.*, pages 427–436. Springer Verlag, 1992.

[76] J. H. Friedman. Regularized discriminant analysis. *J. of the Am. Stat. Assoc.*, 84:165–175, 1989.

[77] G. Galilei. *Discorsi e dimostrazioni matematiche, informo a due nuoue scienze attenti alla mecanica i movimenti locali.* appresso gli Elsevirii; Opere VIII. (2), 1638.

[78] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classication of silhouettes. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 21(12):1312–1328, 1999.

[79] D. Geiger and F. Girosi. Parallel and deterministic algorithms from mrfs: Surface reconstruction and integration. In O. Faugeras, editor, *First Europ. Conf. on Comp. Vision*, volume 427 of *LNCS*, pages 89–98, Antibes, France, April 23–27 1990. Springer Verlag.

[80] S. Geman. Sieves for nonparametric estimation of densities and regressions. Technical Report 99, Division of Applied Math., Brown Univ., Providence, RI, 1981.

[81] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 6(6):721–741, 1984.

[82] S. Geman and C.-R. Hwang. Nonparametric maximum likelihood estimation by the method of sieves. *Annals of Statistics*, 10:401–414, 1982.

[83] C. Goodall. Procrustes methods in the statistical analysis of shape. *J. Roy. Statist. Soc., Ser. B.*, 53(2):285–339, 1991.

[84] J. C. Gower. Generalized procrustes analysis. *Psychometrica*, 40:33–50, 1975.

[85] R. E. Graham. Snow removal: a noise-stripping process for picture signals. *IRE Transactions on Information Theory*, 8(2):129–144, 1962.

[86] G. H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, C–21:195–201, 1972.

[87] U. Grenander. *Lectures in Pattern Theory.* Springer, Berlin, 1976.

[88] U. Grenander, Y. Chow, and D.M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes.* Springer, New York, 1991.

[89] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, 2001.

[90] T. Heap and D. Hogg. Automated pivot location for the cartesian-polar hybrid point distribution model. In *Brit. Mach. Vis. Conf.*, pages 97–106, Edinburgh, UK, Sept. 1996.

[91] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *Brit. Mach. Vis. Conf.*, Colchester, UK, 1997.

[92] H. von Helmholtz. *Handbuch der Physiologischen Optik.* Leopold Voss, Hamburg, 1909.

[93] A. H. Hill and C. J. Taylor. Model based image interpretation using genetic algorithms. *Image and Vision Computing*, 10:295–300, 1992.

[94] J. Hornegger, H. Niemann, and R Risack. Appearance-based object recognition using optimal feature transforms. *Patt. Recog.*, 33(2):209–224, 2000.

[95] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, 23(2):368–388, April 1976.

[96] J. Hoschek and D. Lasser. *Fundamentals of computer aided geometric design.* A.K. Peters, Wellesley, MA, 1993.

[97] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[98] P. J. Huber. *Robust statistics.* Wiley, 1981.

[99] T. Iijima. Basic theory on normalization of pattern (in case of typical one–dimensional pattern). *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962 (in Japanese).

[100] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123–134, 1963 (in English).

[101] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 23:253–258, 1925.

[102] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. of Comp. Vis.*, 1(4):321–331, 1988.

[103] D.G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9:428–430, 1977.

[104] J. T. Kent. The complex Bingham distribution and shape analysis. *J. Roy. Statist. Soc., Ser. B.*, 56:285–299, 1994.

[105] C. Kervrann. *Modèles statistiques pour la segmentation et le suivi de structures déformables bidimensionnelles dans une séquence d'images.* PhD thesis, Université de Rennes I, France, 1995.

[106] C. Kervrann and F. Heitz. A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60:173–195, 5 1998.

[107] C. Kervrann and F. Heitz. Statistical deformable model-based segmentation of image motion. *IEEE Trans. on Image Processing*, 8:583–588, 1999.

[108] S. Kichenassamy, A. Kumar, P. J. Olver, A. Tannenbaum, and A. J. Yezzi. Gradient flows and geometric active contour models. In *Proc. IEEE Internat. Conf. on Comp. Vis.*, pages 810–815, Boston, USA, 1995.

[109] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 12(1):103–108, 1990.

[110] J. J. Koenderink. The structure of images. *Biol. Cybernetics*, 50:363–370, 1984.

[111] G. Koepfler, C. Lopez, and J.-M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM J. Numer. Anal.*, 31(1):282–299, 1994.

[112] T. Kohlberger. Nicht-lineare statistische Repräsentation von Objektformen für die Bildsegmentierung. Diploma thesis (in German), Department of Mathematics and Computer Science, University of Mannheim, Mannheim, Germany, 2001.

[113] P. Kornprobst, R. Deriche, and G. Aubert. Image sequence analysis via partial differential equations. *J. Math. Imag. Vision*, 11(1):5–26, 1999.

[114] F. Kuhl and C. Giardina. Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processessing*, 18:236–258, 1982.

[115] 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.

[116] L. J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 22(10):1185–1190, 2000.

[117] A. Leaci and S. Solimini. Variational problems with a free discontinuity set. In B.M. ter Haar Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, pages 147–154. Kluwer Acad. Publ., Dordrecht, 1994.

[118] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *Int. J. of Comp. Vis.*, 3(1):73–102, 1989.

[119] F. Leitner and P. Cinquin. Complex topology 3d objects segmentation. In *SPIE Conf. on Advances in Intelligent Robotics Systems*, volume 1609, Boston, November 1991.

[120] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. Conf. Computer Vis. and Pattern Recog.*, volume 1, pages 316–323, Hilton Head Island, SC, June 13–15, 2000.

[121] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 17(2):158–175, 1995.

[122] C. Mantegazza. *Su Alcune Definizioni Deboli di Curvatura per Insiemi Non Orientati*. PhD thesis, Dept. of Mathematics, SNS Pisa, Italy, 1993.

[123] D. Marr. *Vision*. W.H. Freeman and Comp., San Francisco, 1982.

[124] D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond.*, B(207):187–217, 1980.

[125] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processessing*, 1:169–182, 1972.

[126] G. E. Martin. *Transformation Geometry, An Introduction to Symmetry*. Springer, New York, 1982.

[127] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. 5th Int. Conf. on Computer Vision*, pages 840–845, Los Alamitos, California, June 20–23 1995. IEEE Comp. Soc. Press.

[128] E. Memin and P. Perez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. on Im. Proc.*, 7(5):703–719, 1998.

[129] S. Menet, P. Saint-Marc, and G. Medioni. B–snakes: implementation and application to stereo. In *Proc. DARPA Image Underst. Workshop*, pages 720–726, April 6-8 1990.

[130] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London, A*, 209:415–446, 1909.

[131] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proc. IEEE Internat. Conf. on Comp. Vis.*, pages 786–793, 1995.

[132] G. H. Montaneri. On the optimal detection of curves in noisy pictures. *Comm. Assoc. Comp. Mach.*, 14:335–345, 1971.

[133] J.-M. Morel and S. Solimini. Segmentation of images by variational methods: a constructive approach. *Revista Matematica de la Universidad Complutense de Madrid*, 1(1,2,3):169–182, 1988.

[134] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhäuser, Boston, 1995.

[135] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *CVPR*, pages 22–26, 1985.

[136] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.

[137] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

[138] M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 14(8):826–833, 1992.

[139] N. Nordström. Biased anisotropic diffusion - a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327, 1990.

[140] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. of Visual Commun. and Image Repr.*, 6(4):348–365, 1995.

[141] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Proc.*, 66:143–155, 1998.

[142] S. Osher and L. I. Rudin. Feature–oriented image enhancement using shock filters. *SIAM J. Numer. Analysis*, 27:919–940, 1990.

[143] S. J. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988.

[144] E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[145] P. Perona and J. Malik. Scale-space and edge-detection. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 12(7):629–639, 1990.

[146] E. Persoon and K.-S. Fu. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3):170–179, 1977.

[147] S. Petitjean, J. Ponce, and D. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *Int. J. of Comp. Vis.*, 9:231–255, 1992.

[148] M. Pfeiffer and M. Pandit. A parametrical description of plane curves using wavelet descriptors. In *Proc. IEEE Int. Conf. on Acoustics Speech and Signal Processing*, volume 4, pages 2531–2534, 1995.

[149] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[150] S. Romdhani, S. Gong, and A. Psarrou. A multi-view non-linear active shape model using kernel pca. In T. Pridmore and D. Elliman, editors, *Proc of the Brit. Mach. Vis. Conf.*, volume 2, pages 483–492, Nottingham, UK, Sep 1999. BMVA Press.

[151] F. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[152] A. Rosenfeld and A. C. Kak. *Digital picture processing, Computer Science and Applied Mathematics*. Academic Press, New York, 2nd edition edition, 1982.

[153] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Trans. Comput.*, 20:562–569, 1971.

[154] M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden et al., editors, *Proc. of the Europ. Conf. on Comp. Vis.*, volume 2351 of *LNCS*, pages 78–92, Copenhagen, May 2002. Springer, Berlin.

[155] S. Roweis. EM algorithms for PCA and SPCA. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 626–632, Cambridge, MA, 1998. MIT Press.

[156] G. Sapiro. Vector self snakes. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 477–480, Lausanne, 1996.

[157] C. Schnörr. Computation of discontinuous optical flow by domain decomposition and shape optimization. *Int. J. of Comp. Vis.*, 8(2):153–165, 1992.

[158] C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals. In *12th Int. Conf. on Pattern Recognition*, Jerusalem, Israel, Oct 9-13 1994.

[159] C. Schnörr. Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals. *J. Math. Imag. Vision*, 4:189–198, 1994.

[160] C. Schnörr and W. Peckar. Motion-based identification of deformable templates. In R. Šára V. Hlaváč, editor, *Proc. 6th Int. Conf. on Computer Analysis of Images and Patterns (CAIP '95)*, volume 970 of *LNCS*, pages 122–129, Prague, Czech Republic, Sept. 6-8 1995. Springer.

[161] B. Schölkopf. *Support Vector Learning*. Oldenbourg, München, 1997.

[162] B. Schölkopf, S. Mika, Smola A., G. Rätsch, and Müller K.-R. Kernel PCA pattern reconstruction via approximate pre-images. In L. Niklasson, M. Boden, and T. Ziemke, editors, *ICANN*, pages 147–152, Berlin, Germany, 1998. Springer.

[163] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space vs. feature space in kernel–based methods. *IEEE Trans. Neur. Networks*, 10(5):1000–1017, 1999.

[164] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[165] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. Wiley, New York, 1992.

[166] B. W. Silverman. Choosing the window width when estimating a density. *Biometrika*, 65:1–11, 1978.

[167] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1992.

[168] P. Sozou, T. Cootes, C. Taylor, and E. Di Mauro. Non-linear generalization of point distribution models using polynomial regression. In E. Hancock, editor, *Proc of the 5th Brit. Mach. Vis. Conf.*, pages 397–406, 1994.

[169] P. Sozou, T. Cootes, C. Taylor, and E. Di Mauro. Non-linear point distribution modelling using a multi-layer perceptron. In D. Pycock, editor, *Proc of the 6th Brit. Mach. Vis. Conf.*, pages 107–116, 1995.

[170] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 14(11):1061–1075, 1992.

[171] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Comp. Vis., Graph., and Imag. Proc.*, 24:52–96, 1983.

[172] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 8(4):413–424, 1986.

[173] D. W. Thompson. *On Growth and Form.* Cambridge University Press, Cambridge, 1917.

[174] A. N. Tikhonov and V. Y. Arsenin. *Solution of ill–posed problems.* V.H. Winston, Washington, D.C., 1977.

[175] M. E. Tipping. Sparse kernel principal component analysis. In *Advances in Neural Information Processing Systems 13*, Vancouver, Dec. 2001.

[176] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Technical Report Woe-19, Neural Computing Research Group, Aston University, UK, 1997.

[177] F. Tischhäuser. Development of a multigrid algorithm for diffusion snakes. Diploma thesis (in German), Department of Mathematics and Computer Science, University of Mannheim, Mannheim, Germany, 2001.

[178] Z. W. Tu and S. C. Zhu. Prior learning and Gibbs reaction–diffusion. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(5), 2002. To appear.

[179] C. J. Twining and C. J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In T. Cootes and C. Taylor, editors, *Proc of the Brit. Mach. Vis. Conf.*, pages 23–32, 2001.

[180] V. N. Vapnik. *The nature of statistical learning theory.* Springer, 1995.

[181] T. J. Wagner. Nonparametric estimates of probability densities. *IEEE Trans. on Inform. Theory*, 21:438–440, 1975.

[182] K. Wang. *Affine–invariant moment method of three–dimensional object indentification.* PhD thesis, Syracuse University, Syracuse, 1977.

[183] Y. Wang and L. H. Staib. Boundary finding with correspondence using statistical shape models. In *Proc. Conf. Computer Vis. and Pattern Recog.*, pages 338–345, Santa Barbara, California, June 1998.

[184] J. Weickert. *Anisotropic diffusion in image processing.* Teubner, Stuttgart, 1998.

[185] J. Weickert. Applications of nonlinear diffusion filtering in image processing and computer vision. *Acta Mathematica Universitatis Comenianae*, LXX(1):33–50, 2001.

[186] J. Weickert and T. Brox. Diffusion and regularization of vector– and matrix–valued images. Technical Report 58, Dept. of Mathematics, Univ. of Saarbrücken, Germany, 2001.

[187] J. Weickert, S. Ishikawa, and A. Imiya. On the history of Gaussian scale–space axiomatics. In J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors, *Gaussian scale–space theory*, pages 45–59. Kluwer, Dordrecht, 1997.

[188] J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in PDE–based computation of image motion. *Int. J. of Comp. Vis.*, 45(3):245–264, 2001.

[189] R. Weiss and M. Boldt. Geometric grouping applied to straight lines. In *Proc. IEEE Conf. on Comp. Vision Patt. Recog.*, pages 489–495, Miami Beach, Florida, 1986.

[190] M. Werman and D. Weinshall. Similarity and affine invariant distances between 2d point sets. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 17(8):810–814, 1995.

[191] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992.

[192] A. Witkin. Scale–space filtering. In *Proc. of IJCAI*, pages 1019–1021, Karlsruhe, 1983.

[193] P. Wunsch and A. Laine. Wavelet descriptors for multiresolution recognition of handprinted characters. *Patt. Recog.*, 28(8):1237–1249, 1995.

[194] M.-H. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 37–40, Vancouver, CA, Sept. 2000.

[195] A. Yezzi, S. Soatto, A. Tsai, and A. Willsky. The Mumford–Shah functional: From segmentation to stereo. *Mathematics and Multimedia*, 2002. To appear.

[196] L. Younes. Optimal matching between shapes via elastic deformations. *Image and Vision Computing*, 17:381–389, 1999.

[197] A. Yuille. Generalized deformable models, statistical physics, and matching problems. *Neural Comp.*, 2:1–24, 1990.

[198] A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille, editors, *Active Vision*, pages 21–38. MIT Press, 1992.

[199] A. Yuille and T. Poggio. Scaling theorems for zero–crossings. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 8(1):15–25, 1986.

[200] N. J. Zabusky and E. A. Overman II. Tangential regularization of contour dynamical algorithms. *J. of Computational Physics*, 52(2):351–374, 1983.

[201] C. T. Zahn and R. C. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C–21:269–281, 1972.

[202] P. M. de Zeeuw. Matrix–dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comp. Appl. Math.*, 33:1–27, 1990.

[203] S. C. Zhu and D. Mumford. Prior learning and Gibbs reaction–diffusion. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 19(11):1236–1250, 1997.

[204] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 18(9):884–900, 1996.