Making Deformable Template Models Operational

Rune Fisker

Lyngby 2000
IMM-PHD-2000-77

IMM

ISSN 0909-3192

Printed by IMM, Technical University of Denmark

Preface

This thesis has been prepared at the Section for Image Analysis, Department of Mathematical Modelling, Technical University of Denmark. It is a partial fulfillment of the requirement for the degree of Ph.D. in engineering.

The thesis is within the field of image processing and computer vision. The subject of the thesis is deformable template models with special focus on the common difficulties, which are related to these type of models.

Reading this thesis requires a basic knowledge of image processing, statistics and optimization.

Acknowledgement

I'm grateful to a large number of persons for help, fruitful discussions and great company. First of all I would like to thank my supervisors Jens Michael Carstensen and Knut Conradsen for their supervision.

I would also like to thank previous and present members of the section for image analysis especially Lars Pedersen, Klaus B. Hilger, Rasmus Larsen, Per R. Andresen, Claus Gramkow, Bjarne Ersbøll, Nette Schultz, Allan A. Nielsen and Johan Dore. I also had excellent discussion with Mikkel B. Stegmann and Henrik Aanæs, who I partly supervised during their master projects.

I'm also in debt to Anil K. Jain and the rest of the people at the PRIP lab for letting me stay with them at Michigan State University, where I had an inspiring visit. I would also like to thank Ingeniør Valdemar Selmer Trane og hustru Elisa Tranes Foundation for the financial support, which made this visit possible.

I would also like to thank the extern opponents at the defense of this thesis, Dr. Tim Cootes and Professor Milan Sonka, for attending and for interesting discussions on the presented topics.

Many thanks to the secretaries Helle Welling and Mette Eltzholtz for their help and to the librarian Finn K. Christensen, who never complained about obtaining just another article or book for me.

Finally I'm very grateful to my lovely wife Helle and daughter Cecilie who allways supported me and never complained when deadlines made me a less common sight.

Lyngby, November 1, 2000

Rune Fisker

Abstract

Deformable template models are a very popular and powerful tool within the field of image processing and computer vision. This thesis treats this type of models extensively with special focus on handling their common difficulties, i.e. model parameter selection, initialization and optimization. A proper handling of the common difficulties is essential for making the models operational by a non-expert user, which is a requirement for intensifying and commercializing the use of deformable template models.

The thesis is organized as a collection of the most important articles, which has been published during the Ph.D. project. To put these articles into the general context of deformable template models and to pass on an overview of the deformable template model literature, the thesis starts with a compact survey of the deformable template model literature with special focus on representation, model parameter estimation, initialization, optimization and performance measures. The original articles - aligned a bit in notation and corrected from discovered spelling errors and other typos - are enclosed in the appendices.

Compared to the literature one contribution is a general scheme for estimation of the model parameters, which applies a combination of a maximum likelihood and minimum distance criterion. Another contribution is a very fast search based initialization algorithm using a filter interpretation of the likelihood model. These two methods

can be applied to most deformable template models making a non-expert user able to use the model.

A comparative study of a number of optimization algorithms is also reported. In addition a general polygon-based model, an ellipse model and a textile model are proposed and a number of applications have been solved. Finally the Grenander model and the Active Appearance Model have been explored and some extensions are presented.

Publications

This section list the publications produced during the Ph.D. period. For the enclosed articles the corresponding appendix and reference are given. If an enclosed article is cited in the survey the reference is shown in italic, e.g. *[68]*.

Publications related to this thesis

R. Fisker, N. Schultz, N. Duta and J. M. Carstensen, *The Grenander Deformable Template Model: A General Scheme*, Submitted (Appendix G and *[70]*).

M. B. Stegmann, R. Fisker, and B. Ersbøll, *Extending and applying Active Appearance Models for automated, high precision segmentation in different image modalities*, submitted (Appendix D and *[165]*).

R. Fisker, J. M. Carstensen, M. F. Hansen, F. Bödker and S. Mørup , *Estimation of Nanoparticle Size Distributions by Image Analysis*, To appear in Journal of Nanoparticle Research (Appendix A and *[67]*).

M. B. Stegmann, R. Fisker, B. Ersbøll, H. H. Thodberg and L. Hyldstrup, *Active Appearance Models: Theory and Cases*, Proc. 9th Danish Conf. on Pattern Recognition and Image Analysis, 49-57.

R. Fisker, N. Schultz, N. Duta and J. M. Carstensen, *A General Scheme for Training and Optimization of the Grenander Deformable Template Model.* Proc. Conf. on Computer Vision and Pattern Recognition, 698-705, Hilton Head Island, USA, 2000

R. Fisker and J.M. Carstensen, *A General Polygon-based Deformable Model for Object Recognition*, Proc. 8th Danish Conf. on Pattern Recognition and Image Analysis, 28-35, Copenhagen, Denmark, 1999 (Appendix C and *[66]*).

R. Fisker, J.M. Carstensen and K. Madsen, *Initialization and Optimization of Deformable Models*, Proc. 11th Scandinavian Conf. on Image Analysis, 295-302, Greenland, 1999 (Appendix F and *[68]*).

J. M. Carstensen, R. Fisker, N. Schultz and T. C. Dörge, *Structural Inference Using Deformable Models*, Machine Vision and Advanced Image Processing in Remote Sensing, 61-71, Springer, 1999

R. Fisker and J.M. Carstensen, *On Parameter Estimation in Deformable Models*, Proc. 14th International Conf. on Pattern Recognition, 763-766, Brisbane, Australia, 1998 (Appendix E and *[65]*).

R. Fisker and J.M. Carstensen, *Automated Visual Inspection of Textile*, Proc. 10th Scandinavian Conf. on Image Analysis, 173-179, Lappeenranta, Finland, 1997 (Appendix B and *[64]*).

R. Fisker and J.M. Carstensen, *Automatisk visuel kvalitetsanalyse af tekstil* (in Danish), Proc. 5th Danish Conf. on Pattern Recognition and Image Analysis, 28-35, Copenhagen, Denmark, 1996.

Publications not related to this thesis

H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen, *Robust Factorization*, submitted.

H. Aanæs, R. Fisker, and J. M. Carstensen, *Robust Structure and Motion*, Proc. 9th Danish Conf. on Pattern Recognition and Image Analysis, Ålborg, Denmark, 1-9, 2000.

R. Fisker, H.F. Poulsen, J. Schou, J.M. Carstensen and S. Garbe, *Use of Image Processing Tools for Texture Analysis of High Energy X-ray Synchrotron Data*, Journal of Applied Crystallography, 31, 647-653, 1998.

Technical reports

M. B. Stegmann, R. Fisker and B. Ersboell, *On properties of Active Shape Models*, Technical Report IMM-Rep-2000-12, Department of Mathematical Modelling, Technical University of Denmark, 2000.

Notation

The notation should be quite consistent throughout the thesis. Bold lowercase letters denotes vectors, $\boldsymbol{x} = (x_0, ..., x_{n-1})$, and bold uppercase letters indicates matrices, $\boldsymbol{X}$.

λ	regularization constant.
Ω_x	parameter space of x.
$P(\boldsymbol{v})$	prior model/distribution.
$P(\boldsymbol{v}\|\boldsymbol{y})$	posterior model/distribution.
$P(\boldsymbol{y}\|\boldsymbol{v})$	likelihood or observation model/distribution.
Σ	covariance matrix.
θ	model parameters.
$\boldsymbol{v}$	template parameters.
$\boldsymbol{v}^t$	training sample of template parameters.
$\hat{\boldsymbol{v}}$	MAP estimate of the template parameters.
$U(\boldsymbol{v}, \boldsymbol{y})$	energy term of the posterior distribution.
$\boldsymbol{y}$	an image.
z	normalizing constant.
$\boldsymbol{X}'$	transposed of a matrix $\boldsymbol{X}$

Contents

Chapter 1

Introduction

The family of deformable template models has been presented under many different names, where the best known probably are Snakes, Active Contour models, Active Shape models, deformable models and deformable templates. A general definition of deformable template models could be:

> *A deformable template model can be characterized as a model, which under an implicit or explicit optimization criterion deforms the shape to match a known type of object in an image.*

In most cases the core task of the deformable template model is to perform segmentation of a known type of object, but the models can also be applied to more general tasks, e.g. image retrieval [16, 102, 104] and tracking [18, 99, 145, 146, 171].

Even though deformable template models have been a very active research field for more than a decade, a recent review of statistical pattern recognition [104] pinpoints deformable models as a one of the frontiers of pattern recognition.

1.1 Bayesian Framework

The design and use of deformable template models fits neatly into the Bayesian framework for image analysis. We refer to Besag [15] or Mumford [137] for a general introduction to the Bayesian framework for image analysis. In a Bayesian setting, the deformable template model framework can be summarized as:

1. Choose a representation of the object, where the template parameters, $\boldsymbol{v} \in \Omega_v$, direct or under some mapping define the object.

2. Construct a prior probability distribution $P(\boldsymbol{v}|\theta)$, where θ corresponds to the model parameters, which determine the properties of the distribution.

3. Formulate a likelihood or observation model, $P(\boldsymbol{y}|\boldsymbol{v}, \theta)$, which defines the probability of an image, $\boldsymbol{y} \in \Omega_y$, given any particular realization of $\boldsymbol{v}$.

4. Combine the prior distribution and the observation model to the posterior distribution, $P(\boldsymbol{v}|\boldsymbol{y}, \theta)$, by Bayes theorem.

$$P(\boldsymbol{v}|\boldsymbol{y},\theta) = \frac{P(\boldsymbol{v}|\theta)P(\boldsymbol{y}|\boldsymbol{v},\theta)}{P(\boldsymbol{y}|\theta)} \quad (1.1)$$

$$\propto P(\boldsymbol{v}|\theta)P(\boldsymbol{y}|\boldsymbol{v},\theta) \quad (1.2)$$

5. Select or estimate the model parameters, θ.

6. Make inference e.g. by estimation of the maximum a posteriori (MAP):

$$\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y},\theta) \quad (1.3)$$

Note, that θ, for simplicity, is omitted in the distribution functions in most of this thesis as well as in the literature, i.e. $P(\boldsymbol{v}|\boldsymbol{y}) = P(\boldsymbol{v}|\boldsymbol{y},\theta)$ etc.

The posterior distribution can be interpreted as an (explicit) optimization criterion and under the weak assumption that the posterior distribution, $P(\boldsymbol{v}|\boldsymbol{y})$, is Gibbs distributed, the posterior distribution can be expanded to:

$$P(\boldsymbol{v}|\boldsymbol{y}) \quad = \quad \frac{1}{z}\exp\{-U(\boldsymbol{v},\boldsymbol{y})\} \tag{1.4}$$

where $U(\boldsymbol{v},\boldsymbol{y}) : \Omega_v \to \mathbb{R}$ is the energy function and z is a normalizing constant, which ensures a proper statistical distribution, i.e. $\int_{\Omega_v} P(\boldsymbol{v}|\boldsymbol{y})dv = 1$. Due to the dimensionality of Ω_v it is usually infeasible to determine the value of z. In practice it is of little importance, since z is a constant, which make the optimization in (1.3) invariant to the actual value of z.

It is very common to formulate the optimization criteria as an energy function instead of a posterior distribution. Under the assumption that $P(\boldsymbol{v}|\boldsymbol{y})$ is Gibbs distributed and z is constant, maximizing the probability is equivalent to minimizing the energy, since:

$$\begin{aligned}\hat{\boldsymbol{v}} &= \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y},\theta) \\ &= \max_{\boldsymbol{v}} -U(\boldsymbol{v},\boldsymbol{y},\theta) - \log(z) \\ &= \min_{\boldsymbol{v}} U(\boldsymbol{v},\boldsymbol{y},\theta)\end{aligned} \tag{1.5}$$

This shows that *the probabilistic formulation is equivalent to the energy formulation* when applied to MAP estimation.

In general we prefer a Bayesian formulation, because it gives a natural separation of model and image contributions in $P(\boldsymbol{v})$ and $P(\boldsymbol{y}|\boldsymbol{v})$, and it provides the opportunity to simulate and thus visualize the appropriateness of the prior model. Further it provides several ways to make inference in $P(\boldsymbol{v}|\boldsymbol{y})$.

1.2 The Prior and Likelihood Model

The basic principle of the prior model, $P(\boldsymbol{v})$, is to introduce prior knowledge into the model. The attraction of using prior knowledge is simply that it is so hard to make progress without it [18]. One interpretation of the prior model is to see it as a regularizer [147, 148], which regularizes the shape of the template. In fact, the prior model in the Snake [108] (equal to the internal energy) is a generalization of a Tikhonov [173] regularizer [108].

From a statistical point of view the prior model imposes a distribution on the template parameters. This distribution introduces correlation between parameters and governs the overall deformation. Fisker et al. *[65]* demonstrate that in the case of the prior energy being a quadratic term, the prior model usually corresponds to a multi-dimensional Gaussian. As demonstrated later this result has important applications within model parameter estimation as well as model simulation and validation. Using this result the prior model of the snake correspond to a multi-dimensional Gaussian with the mean equal to zero and the inverse covariance equal to a pentadiagonal banded matrix.

The likelihood or observation model, $P(\boldsymbol{y}|\boldsymbol{v})$, is the data driven term. The model defines the interaction between a realization of the template $\boldsymbol{v}$ and an observed image $\boldsymbol{y}$. Basically it defines how well the actual template match the object in the image. Most likelihood models are based on image intensity [23, 41, 35, 85, 86, 167, 155, 189] and/or gradient (edge) information [33, 34, 72, 101, 102, 108, 118, 163, 189], but in principle all kinds of information can be combined e.g. texture, colour or motion.

1.3 Common Difficulties

In general, the current approaches to deformable template models suffers from a number of common difficulties. Jain et al. [101] summarize the difficulties as:

- The user needs to assign weights to different components of the model. The selection of these weights and other parameters determine the success of a deformable template model (Model Parameter Selection).

- The algorithms need a good initialization to give meaningful results, otherwise they get stuck at spurious local minima and thus lead to incorrect results (Initialization).
- The large number of parameters makes the numerical solutions difficult (Optimization).

In short, the common difficulties can be summarized as Model Parameter Selection, Initialization and Optimization. These difficulties are the main reason why most models only can be applied by an expert. This is not satisfactory since a good model should be applicable by a non-expert user. The handling and solution to these difficulties are covered intensively in the following chapters.

1.4 Popular Deformable Template Models

Jain et al. [101, 102] divide deformable models into two groups: Free form and Parametric. Free form deformable templates have no explicit global structure because their prior model contains only local continuity and smoothness constraints. The best known example of free form models are Snakes [108] and Ballons [33, 34]. In parametric deformable models prior knowledge of the global structure is included using a parameterized template. Examples of parametric deformable models are Grenander's model [85], Active Shape model [41], Active Appearance model [35] and Blake's Active Contour [18]. The cited models all have the important property of being general in the sense that they can be applied to an object with an arbitrary shape. A more detailed overview of the properties of the most popular general models is given in table 1.1. The first reference in the table is the "key reference" and Eucl. is an abbreviation for Euclidean parameters (position, scale and rotation).

The largest group of deformable models are formulated and tuned for a specific object. Properly the best-known specific model is the eye model and a mouth model proposed by Yuille et al. [189]. For some problems it is necessary to incorporate specific assumption into the model to be able to solve the problem, but many problems could have been solved directly by one of the general models.

It is outside the scope of this thesis to give a general description of all the models in the literature. For a general survey we refer to the first part of the book on Active Contours by Blake and Isard [18] and the survey papers by McInerney and Terzopoulos [132] and by Jain et al. [101].

1.5 Relationship To Other Methods

Deformable template models have close relations to a number of other computer vision and image processing methods, where the most import are covered in the following sections.

1.5.1 Rigid Template Matching

Early research in template matching concentrated mainly on the simple rigid shape matching, where a prototype template was transformed by simple transformations like translation, rotation and scaling before it was matched to the image. The matching is typically performed by a simple correlation-based method using a filtering approach, see e.g. [2, 162]. This type of matching is still very popular in machine vision applications.

The rigidness of this approach makes it insufficient for objects with larger shape variations, which basically lead to the development of deformable template models. For many models the likelihood model, $P(\boldsymbol{y}|\boldsymbol{v})$, can be interpreted as correlation-based match of a deformed template.

Model	Object Representation	Template Parameters v	Model Parameters θ	Weight Selection	Initialization
Active Appearance Model [35, 40, 61]	Mean shape/bitmap, linear combination of eigenvectors & eucl.	Scalars on eigenvectors & eucl.	Eigenvectors & mean shape/bitmap	No weights	Search
Active Shape Model [38, 40, 41]	Mean shape, linear combination of eigenvectors & eucl.	Scalars on eigenvectors & eucl.	Eigenvectors & mean shape	No weights	No comments
Balloon [33, 34]	Points/FEM	Points/ FEM	Weights	Manual	Manual
Blake's Active Contour [18]	B-Spline & deformation space	Curve shape-vector	Mean shape, shape matrix & weights	No comments	Moments binary image
Grenander's model [85, 86, 114, 155]	Points	Points	Mean shape, gray level stat. & weights	Manual	Heuristic
Jain's model [101, 102, 190]	Deformation of bitmap & eucl.	Deformation parameter & eucl.	Bitmap & weight	Manual	Search
Snake [108]	Points	Points	Weights	Manual	Manual
Staib & Duncan's Model [163]	Elliptic Fourier descriptors	Elliptic Fourier descriptors	Fourier means & weight	Manual	Mode of prior

Table 1.1: Properties of popular 2D deformable template models

1.5.2 Hough Transform

The Hough transform was first proposed by Hough [95]. Since then, numerous papers have improved the Hough transform, where the (ρ, θ)-representation of lines [58] and the generalization to arbitrary shapes for any scale and any rotation [11] are some of the most influential. As demonstrated by Stockman and Agrawala [166], the Hough transform is simply an efficient implementation of template matching. For more a extensive review we refer to the survey papers [98, 122].

In a deformable model setting the Hough transform can be interpreted as a deformable model with a uniform prior, where the initialization is performed by an intensive but efficient search and the optimization is omitted.

1.5.3 Registration

Image registration requires finding an optimal transform between an image pair, the source and target image [123]. For a review of image registration, we refer to one of the survey papers [7, 21, 123, 129, 177, 183].

Matching a parametric deformable model to an object in an image can in general be interpreted as a registration of a prototype template to the object. Still there is difference in the actual objective. In registration the requirement is an exact correspondence between homologous points, where deformable models are focused on segmentation rather than necessarily finding the exact correspondence. Another difference is the actual matching. Traditionally registration is performed between similar feature types, e.g. pairs of images with the same or different modality or pairs of landmark sets. Whereas most deformable models match an object model – with a non-image representation – to an image.

Still there exist a number of methods, which can be classified as both a registration method and a deformable template model, e.g. elastic models like [9, 10, 49] and the viscous fluid model [20, 29, 30].

1.5.4 Markov Random Fields

Markov Random Fields (MRF) provide a convenient and consistent way of modeling spatial context and stochastic interaction among entities such as image pixels and other spatially correlated features. For an elaborate treatment of MRFs we refer to [125].

From a general point MRF's and deformable template models apply a very similar optimization based framework and both fit neatly into the Bayesian setting. Li [125] also argue that many deformable templates models can be interpreted as a high level MRF having irregular sites with discrete labels, where each site indexes an image feature such as a point or a line segment.

Despite these similarities the literature is quite clear on what is classified as a MRF and a deformable template model, respectively. The main difference is what is modeled. Deformable templates models is based on an object model whereas MRFs model the full image or the interaction between objects.

Chapter 2

Representation

Choosing object representation is one of the most important choices in the design of a deformable template model. The most popular representations are summarized bellow. We have chosen to categorize the representations in three main groups: Curves, Orthogonal basis and Image Templates.

2.1 Curves

2.1.1 Labeled Points and Vertices

One of the first and most popular representation is a labeled set of points with connectivity information. This representation is equivalent with a vertices and edges representation. In the simple case of one single open or closed contour neighbor points are assumed to be connected. In this case the point representation corresponds to a polygon or a linear spline. Numerous authors have used the point representation, e.g. [22, 33, 57, 85, 108, 118].

2.1.2 B-splines

Another popular representation is B-splines, which enable a continues curve description, see e.g. [18] for an introduction. Compared to a point representation B-splines also have the advantage of a low-dimensional parameter space, since a similar shape represented is obtained by a few control points compared to a large number of points. Another advantage is the "built-in" smoothness. Realization of deformable models using B-splines was developed by Cipolla and Blake [31], Menet et al. [133] and Hinton et al. [92]. Other examples of the use of B-splines representation are given in [18, 99, 113].

2.1.3 Level Sets

Another powerful representation is based on the elegant level sets [142]. See [159] for an introduction. Compared to the other representations level sets have the advantage of allowing automatic merging and splitting of the initial contour. In the context of deformable models level sets were simultaneously proposed by Caselles et al. [25] and Malladi et al. [130]. Since then level sets have received much attention and a number of models have been based on this approach, see e.g. [26, 111, 124, 144, 143, 186, 187].

2.1.4 Other Analytic Curves

Beside the mentioned curves, numerous other sets of analytic curves have been applied to represent the object, e.g. a set of parabolic curves and circles. Examples of representation based on other sets of curves are [67, 119, 189].

2.2 Orthogonal basis

2.2.1 Fourier Descriptors

Fourier descriptors represent the object on an orthogonal basis, where the usual basis function is the sinusoids, i.e. trigonometric functions. On a sinusoidal basis the representation correspond to the elliptic Fourier descriptors [78, 116, 127, 163], where the Fourier descriptors are localized in frequency. In the context of deformable template models Descriptors were introduced by Scott [158] and Staib and Duncan [163].

2.2.2 Principal Component Analysis (PCA)

Another representation which applies an orthogonal basis is the Point Distribution Model proposed by Cootes et al. [36, 41]. In practice, the object is represented by the mean shape of a training set and a linear combination of the most important eigenmodes of the shape variation from this mean. This representation has close connections to the general Shape Statistic, see e.g [56]. The Point Distribution Model plays an important role in the popular Active Shape Model [41] and has been extended with texture in the novel Active Appearance Model [35]. Numerous other models like [60, 126, 139] have been based on this representation.

2.2.3 Wavelets Descriptors

A third orthogonal basis is the wavelet transform [47, 48], see e.g. [131, 168] for a general introduction to wavelets. Wavelets are localized both in in space and in frequency (scale), since they are defined as dilated and translated version of the basis or mother wavelet. Wavelets descriptors are less popular than the Fourier descriptors and the Point Distribution Model, but there are still examples like

[139, 188, 190], where [139, 188] apply the Daubechies wavelet [47, 48] and [190] applies the B-spline wavelet [175, 176], respectively. Note the comparison of shape models based on the point distribution model, Fourier descriptors and wavelets in [139]. The previous examples all applies wavelets for the shape representation, but wavelets have also been used to make a compact texture representation [184] in Active Appearance Models [35].

Note that the orthogonal basis usually apply a reduced or truncated parameter space where only the most important modes and descriptors are used.

2.3 Image Templates

Another popular representation is a prototype image template. The prototype image is then deformed under a similarity measure to match a new object in an image. Most of these models can also be classified as registration methods (see discussion in section 1.5.3). Typically the template is of the same type as the input image but edges templates have also been used, see [102]. There is a rich collection of examples of this representation, where some of the best known are [5, 9, 10, 29, 30, 100, 156].

2.4 Conclusion and Discussion

The central question is now which representation is the best? Unfortunately there is no easy answer to this question, since it dependents on the rest of the model and the actual problem. Still there is a number of properties, which from a general point of view are desirable for the ideal model:

- General.
  The representation should be able to model an arbitrary object.

- Low redundancy and dimensionality.
  A low dimensional representation with little redundancy improves the computational efficiency and make the optimization easier and more robust.
- Linear parameterization.
  Restriction to linear parameterization has certain advantages in simplifying fitting algorithms and avoiding problems with local minima [18].
- Orthogonal basis
  In general, an orthogonal basis is desirable because it makes the parameters distinct. This makes the coefficients determination easier and avoids redundancy [163].
- Shape regularization.
  Commonly regularization of the shape is obtained by the prior model and the representation. In many cases the regularization of the shape is what makes the deformable model successful, and it is an advantage if the representation can be used as a regularizer.
- Automatic splitting and merging.
  Primarily in the context of free form deformable model automatic splitting and merging is a desirable property.

These properties favors some models, but still there is no superior representation.

Chapter 3

Model Parameter Estimation

Model parameters play an important role in determining the properties of the posterior distribution. Commonly the model parameters are selected before the model is applied and these are kept constant. Examples of model parameters are the mean and variance in a Gaussian distribution and the weight parameters in the snake, which determine the relative influence of the different terms in the energy function.

Most deformable models contain model parameters, e.g. weight parameters. With a few exceptions these parameters are selected manually by an expert (c.f. Table 1.1). From a general point of view this is not acceptable since the goal is a framework, which can be applied by a non-expert user. Methods applied to automatic model parameter selection in deformable template models are described below.

3.1 Minimax

Gennert and Yuille [77] propose a general unsupervised technique for determination of model parameters for multiple objective function optimization based on the minimax principle. Recall $\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y})$ $= \min_{\boldsymbol{v}} U(\boldsymbol{v},\boldsymbol{y})$. The principle is to determine the θ that maximizes $U(\hat{\boldsymbol{v}},\boldsymbol{y},\theta)$ when $U(\boldsymbol{v},\boldsymbol{y},\theta)$ has been minimized over $\boldsymbol{v}$. More formally the minimax criterion is defined as [77]:

$$\hat{\theta} = \max_{\theta} \min_{v} U(\boldsymbol{v},\boldsymbol{y},\theta) \tag{3.1}$$

The minimax principle is a very conservative estimate, since the optimal model parameters correspond to the worst case energy when the deformable template model has been optimized. This may seem at first far from optimal, but if one recalls that $U(\boldsymbol{v},\boldsymbol{y},\theta)$ has been minimized over $\boldsymbol{v}$, it will be seen that the maximization over θ does make sense, while avoiding the problems of θ being excessively low or high [77]. The minimax principle has been applied in [117, 118] to estimate the regularization parameter in their g-snake. Note that the minimax principle is based on the assumption that $\min_v U(\boldsymbol{v},\boldsymbol{y},\theta)$ is convex with respect to θ, which is often not the case *[65]*.

3.2 Maximum Likelihood

Assume a ground truth of n samples, $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{n-1}^t$, and their corresponding images, $\boldsymbol{y}_0^t, ..., \boldsymbol{y}_{n-1}^t$, has been supplied. Given this training set the model parameters can – in theory – be estimated by the Maximum Likelihood (ML) estimate:

$$\hat{\theta} = \max_{\theta} L(\theta, \boldsymbol{v}_0^t, ..., \boldsymbol{v}_{n-1}^t, \boldsymbol{y}_0^t, ..., \boldsymbol{y}_{n-1}^t) \tag{3.2}$$

where $L(\theta, \boldsymbol{v}_0^t, ..., \boldsymbol{v}_{n-1}^t, \boldsymbol{y}_0^t, ..., \boldsymbol{y}_{n-1}^t)$ is the likelihood function. If $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{n-1}^t$ and $\boldsymbol{y}_0^t, ..., \boldsymbol{y}_{n-1}^t$ are assumed to be stochastically independent the Maximum Likelihood estimate becomes:

$$\hat{\theta} = \max_{\theta} \prod_{i=0}^{n-1} P(\boldsymbol{v}_i^t|\boldsymbol{y}_i^t,\theta) \tag{3.3}$$

In *[65]* and [96, 109]*[70]* the ML estimator has been used to estimate a subset of the model parameters corresponding to the prior model and all the model parameters, respectively.

Unfortunately $\prod_{i=0}^{n-1} P(\boldsymbol{v_i}|\boldsymbol{y_i},\theta)$ is not guaranteed to be a convex function with respect to all elements in θ. In fact it is very rare that the likelihood model, $P(\boldsymbol{y}|\boldsymbol{v},\theta_l)$, is convex with respect to the model parameters, θ_l, which relates to the likelihood model. Whereas the prior model, $P(\boldsymbol{v}|\theta_p)$, usually can be rewritten to become convex with respect to the prior model parameters, θ_p. In fact most prior models are born non-convex, but if the corresponding energy function only contains a quadratic term, Fisker and Carstensen *[65]* show that the prior model usually corresponds to a multivariate Gaussian, which is convex with respect to θ_p. The importance of this result can be demonstrated by a univariate Gaussian, where the energy correspond to $U(x) = \frac{(x-\mu)^2}{2\sigma^2}$, which is non-convex w.r.t. σ^2. Since $U(x)$ corresponds to a Gaussian the normalizing constant, z, is known and the energy can be rewritten to $U(x) = \frac{(x-\mu)^2}{2\sigma^2} - \log(\sqrt{2\pi\sigma^2})$, which is convex w.r.t. σ^2.

In general the convexity of $\prod_{i=0}^{n-1} P(\boldsymbol{v_i}|\boldsymbol{y_i},\theta)$ w.r.t. θ_j is closely related to whether the parameter, θ_i, corresponds to a part of the posterior distribution, which isolated corresponds to a proper distribution. If θ_j correspond to a proper distribution, $\prod_{i=0}^{n-1} P(\boldsymbol{v_i}|\boldsymbol{y_i},\theta)$ is usually convex. Whereas $\prod_{i=0}^{n-1} P(\boldsymbol{v_i}|\boldsymbol{y_i},\theta)$ usually is non-convex if θ_j correspond to an improper distribution.

Note that the ML criterion is most sensible for parametric deformable template models, since this type of models favor a known global structure, which can be interpreted as the mean of the prior.

3.3 Minimum Distance

It is infeasible to perform a ML estimation of the model parameters, θ_i, which correspond to a non-convex likelihood function. In this case we propose to use a minimum distance criterion (same concept but an improved criterion compared to *[65]*):

$$\hat{\theta} = \min_{\theta} D(\boldsymbol{V}^t, \hat{\boldsymbol{V}}(\theta)) \tag{3.4}$$

where $D(\boldsymbol{V}^t, \hat{\boldsymbol{V}}(\theta))$ is the distance between $\boldsymbol{V}^t = \boldsymbol{v}_0^t, ..., \boldsymbol{v}_{n-1}^t$ and $\hat{\boldsymbol{V}}(\theta) = \hat{\boldsymbol{v}}_0(\theta), ..., \hat{\boldsymbol{v}}_{n-1}(\theta)$ for $\hat{\boldsymbol{v}}_i(\theta) = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y}^t, \theta)$ being the MAP estimate obtained by running the full initialization and optimization scheme with the actual value of θ. In fact this is a very reasonable criterion, since it corresponds to the overall goal of performing the best segmentation, i.e. the segmentation with the smallest possible distance to the ground truth.

3.3.1 Distance Functions and Performance measures

To apply the minimum distance criterion the distance has to be defined. Unfortunately there is no common standard for performance evaluation of deformable template models, hence there is no common function to evaluate the distance between the ground truth and the segmentation. In fact this is very annoying since it is impossible to compare the performance of different models.

The most popular distance measures used with deformable models are summarized bellow (θ is omitted for simplicity). Note that $D(\boldsymbol{V}^t, \hat{\boldsymbol{V}})$ and $D(\boldsymbol{v}^t, \hat{\boldsymbol{v}})$ correspond to the distance between two training sets and the distance between two samples, respectively. Most of the distance measures assume two finite point sets, $\boldsymbol{U}^t = (\boldsymbol{u}_0^t, ..., \boldsymbol{u}_{p-1}^t)$ and $\hat{\boldsymbol{U}} = (\hat{\boldsymbol{u}}_0, ..., \hat{\boldsymbol{u}}_{q-1})$ can be obtained given $\boldsymbol{v}^t$ and $\hat{\boldsymbol{v}}$, respectively. Usually these points lay on the boundary.

- **Template parameter errors**
  A simple distance measure used in *[65]* is the sum of weighted square errors between the template parameters:

$$D_{sq}(\boldsymbol{V}^t, \hat{\boldsymbol{V}}) = \frac{1}{nl} \sum_{j=0}^{n-1} (\boldsymbol{v}_j^t - \hat{\boldsymbol{v}}_j)' \boldsymbol{C} (\boldsymbol{v}_j^t - \hat{\boldsymbol{v}}_j) \tag{3.5}$$

  where l is the number of parameters in $\boldsymbol{v}$ and $\boldsymbol{C}$ is a symmetric weight matrix. $\boldsymbol{C}$ is usually the identity matrix.

- **Point to point errors**
  Given two finite point sets, $\boldsymbol{U}^t$ and $\hat{\boldsymbol{U}}$, with unknown correspondence and the same number of elements the average point to point error with linear reparameterization reported in [163] is defined as:

$$D_{\overline{ptpt}}(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \min_{0 \le s_0 < k} \frac{1}{k} \sum_{i=0}^{k-1} \|\boldsymbol{u}_i^t - \hat{\boldsymbol{u}}_{i+s_0}\| \tag{3.6}$$

  where $k = p = q$ and s_0 is the offset.

  Another point to point error with known correspondence used in [43] is the root mean square (RMS) point to point error defined as:

$$D_{ptpt}(\boldsymbol{V}^t, \hat{\boldsymbol{V}}) = \sqrt{\frac{1}{nk} \sum_{j=0}^{n-1} \sum_{i=0}^{k-1} \|\boldsymbol{u}_{j,i}^t - \hat{\boldsymbol{u}}_{j,i}\|^2} \tag{3.7}$$

  A third point to point error measure is defined by:

$$D_{ptpt}(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \frac{1}{k} \sum_{i=0}^{k-1} \min_{\boldsymbol{u}_j^t \in \boldsymbol{U}^t} \|\boldsymbol{u}_j^t - \hat{\boldsymbol{u}}_i\| \tag{3.8}$$

  In [60] is reported the mean and standard deviation for this measure for the full test set, $\boldsymbol{V}$. Note this criterion makes no assumption about point correspondence and identical size of the point sets.

- **Undirected Hausdorff distance**
  Given two finite point sets, $\boldsymbol{u}^t$ and $\hat{\boldsymbol{u}}$, the directed Hausdorff distance is defined as [97]:

$$h(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \max_{\boldsymbol{u}_i^t \in \boldsymbol{U}^t} \min_{\hat{\boldsymbol{u}}_j \in \hat{\boldsymbol{U}}} \|\boldsymbol{u}_i^t - \hat{\boldsymbol{u}}_j\| \tag{3.9}$$

  The undirected Hausdorff distance is then defined as [97]:

$$D_H(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \max(h(\boldsymbol{u}^t, \hat{\boldsymbol{u}}), h(\hat{\boldsymbol{u}}, \boldsymbol{u}^t)) \tag{3.10}$$

  The average and standard deviation of the undirected Hausdorff distance for the full set, $\boldsymbol{V}^t$, is reported in [60].

- **Undirected partial Hausdorff distance**
  Given two finite point sets, $\boldsymbol{u}^t$ and $\hat{\boldsymbol{u}}$, the partial directed Hausdorff distance is defined as [124]:

$$h_K(\boldsymbol{u}^t, \hat{\boldsymbol{u}}) = \underset{\boldsymbol{u}_i^t \in \boldsymbol{U}^t}{K^{th}} \min_{\hat{\boldsymbol{u}}_j \in \hat{\boldsymbol{U}}} \|\boldsymbol{u}_i^t - \hat{\boldsymbol{u}}_j\| \tag{3.11}$$

  where K is a quantile of the maximum distance. The undirected distance is then defined using (3.10). This measure is reported in [124] for two different samples and two different quantiles, K.

- **Normal displacement.**
  Given two finite point sets, $\boldsymbol{u}^t$ and $\hat{\boldsymbol{u}}$, with correspondence and connectivity information the squared normal displacement is defined as [18]:

$$D_n(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \frac{1}{k} \sum_{i=0}^{k-1} ((\boldsymbol{u}_i^t - \hat{\boldsymbol{u}}_i) \cdot n(\hat{\boldsymbol{u}}_i))^2 \tag{3.12}$$

  where $n(\boldsymbol{u}_i)$ is the unit normal at $\boldsymbol{u}_i$. This distance measure is extensively used in [18].

- **Point to associated boundary.**
  Given a finite point set, $\hat{\boldsymbol{u}}$, and a piecewise continuous template curve, $r(s, \boldsymbol{v}^t)$, of the ground truth the RMS point to associated boundary error reported in [43] is defined by:

$$D_{pt-b}(\boldsymbol{V}^t, \hat{\boldsymbol{V}}) = \sqrt{\frac{1}{nk}\sum_{j=0}^{n-1}\sum_{i=0}^{k-1} \| \min_{s_i \in \Omega_i} (\hat{\boldsymbol{u}}_{j,i} - r(s_i, \boldsymbol{v}_j^t))\|^2} \tag{3.13}$$

  where Ω_i is the part of the boundary of $r(s, v)$, which is associated with $\boldsymbol{u}_i$.

- **Area error**
  The relative area error is defined as:

$$D_A(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \frac{A(\hat{\boldsymbol{v}})}{A(\boldsymbol{v}^t)} \tag{3.14}$$

  where $A(\boldsymbol{v})$ is the area. In [60] the mean and standard deviation of the relative area is reported.

- **Labeling error**
  The relative labeling error is defined as:

$$D_L(\boldsymbol{v}^t, \hat{\boldsymbol{v}}) = \frac{N_{incor}(\boldsymbol{v}^t, \hat{\boldsymbol{v}})}{N_{Pixels}(\boldsymbol{v}^t)} \tag{3.15}$$

  where N_{incor} is the number of incorrectly labeled pixels and N_{Pixels} is the total number of pixels. The mean and standard deviation of this measure is also given in [60].

- **Texture error**
  Given a texture model, $\boldsymbol{y}(r, c, \hat{\boldsymbol{v}})$, the RMS texture error reported in [43] is defined by:

$$D_t(\boldsymbol{V}^t, \hat{\boldsymbol{V}}) = \sqrt{\frac{1}{n|\Omega_y|}\sum_{j=0}^{n-1}\sum_{(r,c)\in\Omega_y} (\boldsymbol{y}_j(r,c)^t - \boldsymbol{y}(r,c,\hat{\boldsymbol{v}}_j))^2} \tag{3.16}$$

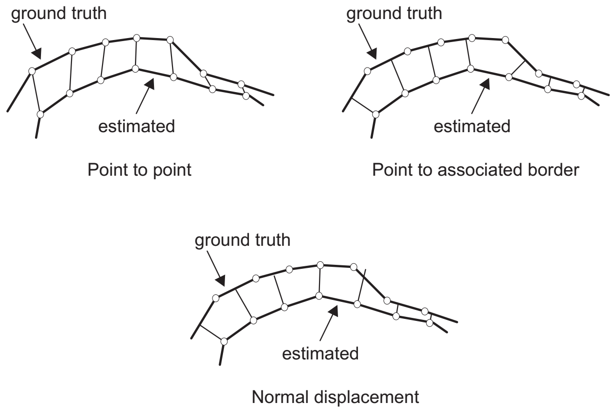


Figure 3.1: Examples of distance measures.

  where $|\Omega_y|$ is the number of members in Ω_y and Ω_y is the set of image coordinates, (r,c), which correspond to the model. Alternative texture metrics can be found in [152].

Some of the distance measures are illustrated in Figure 3.1. Note, that special cases exists where a number of the measures above are identical.

3.3.2 The "Best" Distance Function

It is impossible to select the overall "best" distance function, since it depends on the actual application and model. Still some distance functions seem more attractive than others.

From a general point of view the template parameter error is less attractive, since it is very sensitive to the parameterization and to different types of templates parameters.

Points on the boundary of a deformable model rarely correspond to mathematical or anatomical landmarks. In fact points are allowed to move along the boundary in most applications. If the model is not having a point based representation points need to be created, which is very sensitive to the parameterization. Overall this makes the point to point based measures seem less attractive.

To avoid problems with points moving along the boundary, information of the boundary need to be included. The only measures, which satisfies this is the normal displacement and the point to associated boundary. These criteria basically approximate:

$$D(\hat{\boldsymbol{v}}, \boldsymbol{v}^t) = \frac{1}{L} \int_{s=0}^{1} \|r(s, \hat{\boldsymbol{v}}) - r(g(s), \boldsymbol{v}^t)\|^2 ds \tag{3.17}$$

where $g(s)$ is an reparameterization function defined such that $g(s) = s$ gives the original inherited parameterization, mapping points on the curve $r(s, \hat{\boldsymbol{v}})$ to points on $r(g(s), \boldsymbol{v}^t)$. Unfortunately the true $g(s)$ is impossible to construct. Note that simple numerical approximations of (3.17) only is possible if the error is not squared.

The normal displacement relies on two point sets with established correspondences and the calculation of the normal. This makes the normal displacement seem less attractive than the point to associated boundary. The relative labeling error and relative area error are not very informative due to the to low specificity and the texture error only applies to texture based models, which makes these criteria seem less interesting in the general context.

With respect to the minimum distance estimation of θ the sum of squared point to associated boundary errors seems like the best overall distance function. The squared error is selected since this is the common fitting criterion. In the case of large deviations between the shapes a symmetric or undirected point to associated boundary errors could be considered, e.g. by summing the directed errors.

3.3.3 The "best" Performance Measures

The best distance function for minimum distance estimation is not necessarily the best performance measure since the objective might be different. Still the arguments for the point to associated boundary error are valid, but the sum of squared errors seems harder to interpreted for the user than the error. In fact it seems more relevant to show the full histogram of the point to associated boundary errors or illustrate the average errors on the different positions of the object than obtaining a single number. If a single user friendly number is needed the mean or a proper quantile of the histogram could be obtained. However this is a fundamental problem of model fitting and residual analysis, where there is no unique solution.

3.4 Other Techniques

Larsen et al. [121] propose a two step procedure to select the elasticity parameters of a snake based on upper and lower bounds of the parameters. However this method seems limited to snakes.

Amini et al. [4] discuss the cross validation method proposed by Wahba et al. [46, 181]. This is a popular method for estimation of the regularization parameter in the context of regression and splines, see e.g. [160]. The optimal parameter is found by minimizing the average discrepancy between the data samples and the predicted values using cross validation. For an elaborate treatment we refer to [180]. However the methods are mainly focused on equations of the form $\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int_a^b (f^{(m)}(x))^2 dx$, which makes it less interesting in the context of deformable template models. Another problem is that only a limited group of deformable template models can make a reasonable prediction of a removed template parameter, v_i.

3.5 Minimal Manual Interaction

The manual task of creating training sets is very cumbersome and an automatic algorithm would be preferable. Cootes et al. [37] solve the problem of few training samples by generating artificial samples by a FEM model. However this is basically not different from imposing an artificial covariance structure as done in e.g. the snake. The only difference is how the covariance structure is created.

A large number of semi-automatic methods have been proposed to reduce the needed manual interaction. Most of these methods assume a set of dense sampled object outlines are supplied. Note that it is much easier to create object outlines than actually placing e.g. landmarks. The output from the algorithms are usually a set of subsampled and registered object outlines. Examples of these types of methods are [6, 59, 62, 157, 182].

An alternative strategy is proposed by Fisker et al. *[65, 70]* where the problem is solved by applying the Expectation-Maximization (EM) algorithm [51]. In *[65]* no user interaction is needed whereas one example shape is required in *[70]*.

3.6 Conclusion and Discussion

In this chapter two quite general methods for estimation of the model parameters in deformable models are presented. Both methods can be applied by a non-expert user.

The first method is based on the minimax criterion and has the important feature of not requiring any user interaction. However it still seems very counter intuitive to choose the parameter corresponding to the worst case energy. Another problem is that the minimax criterion rely on an assumption of convexity, which is not always fulfilled.

The second method for parameter estimation is based on a combination of a ML estimate and a minimum distance criterion. The ML criterion should be applied if possible, since it is much easier to optimize than the minimum distance criterion. As a rule of thumb the ML criterion can be applied to estimate parameters, when the parameter is related to a part of the posterior distribution, which isolated corresponds to a proper distribution. This criterion has the drawback of requiring a training set, which can be a cumbersome task to create manually. However a number of methods exists, which reduce or remove the need for manual interaction.

We also review a large number of distance and performance measures. Based on the previous discussion we suggest that the most reasonable overall distance measure is the point to associated border. But it should be stressed that the choice of distance and performance measures is highly dependent on the application and the model.

Note that experimental results for parameter estimation can be found in *[65, 70]*.

Chapter 4

Initialization

To make inference about an object in an image $\boldsymbol{y}$, estimation of the maximum a posteriori (MAP) is performed:

$$\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y}) \tag{4.1}$$

The MAP estimation is with few exceptions carried out using an optimization procedure, which presume an initial configuration of the template parameters. This basically separates the MAP estimation into two steps: initialization and optimization. This chapter is dedicated to initialization and the following to optimization.

The request for the initial configuration is, that it should be close enough to the object in the image to achieve successfully refinement during the optimization. Obviously the initialization is very crucial for the success of the segmentation, since the optimization fails if the initial configuration is not close enough to the object. A general initialization scheme is also essential for a fully automated framework, which can be applied to a new problem by a naive user. Still initialization has received very little attention in the literature and many authors do not even comment on how they initialize.

In the following we classify different initialization strategies into three main groups: Manual, feature based and model based initialization.

Even though manual initialization is very popular, we will not treat this topic further.

4.1 Feature Based Initialization

Feature based initialization extracts some features from the image. These features are then used for initialization of the template. Most feature based methods are ad hoc methods tuned to a specific problem and are of no general interest. Unfortunately this is how most automatic schemes work. Still there exists a few interesting feature based strategies, which are more or less general.

4.1.1 Initialization From Moments

Blake et al. [18] propose to perform the initialization from invariant moments of a binary blob. The binary blob is obtained by thresholding the image or a preprocessed version. A prototype shape is then aligned from these moments with respect to Euclidean or affine similarities. Unfortunately binary blobs are not obtainable in the general case, but initialization from moments still seems like a fast and efficient method for a range of problems.

4.1.2 Texture and Color.

Zhong and Jain [190, 191] propose to use local texture and color features for initialization. From a prototype image of the object local texture and color is extracted. To locate a new object regions with similar color and texture patterns are identified and used for initialization. To speed up the processing texture and color features are directly extracted from the compressed domain of the discrete cosine transform.

4.1.3 Generalized Hough Transform.

Lai and Chin [118] and Garrido and Blanca [72, 73] propose to perform the initialization using the generalized Hough transform [11] and a modified version of the generalized Hough transform, respectively. A prototype shape is supplied as the arbitrary shape to detect. Recall that the generalized Hough transform is invariant with respect to Euclidean transformations. Even though the generalized Hough transform is a powerful method it is not a general initialization method since it relies on gradient information.

4.2 Model Based Initialization.

In model based initialization the full or a part of the posterior model, $P(\boldsymbol{v}|\boldsymbol{y})$, is applied. Staib et al. [163] use the mode of the prior distribution, $P(\boldsymbol{v})$, for initialization, i.e. the configuration of the template parameters which have the highest prior probability - usually the mean shape. Unfortunately this method is only applicable for a few models, since most prior models are invariant to Euclidean transformations.

4.2.1 Search Based Initialization.

The only truly general initialization approach, which we have knowledge about, is the search strategy [61, 102, 164]*[66, 67, 68, 70, 165]*, where a sparse search is performed in the parameter space Ω_v. The static search strategy used by Jain et al. [102] and Fisker et al. *[66, 67, 68, 70]* can be summarized as:

1. Create relevant search configurations $\boldsymbol{v}_0, ..., \boldsymbol{v}_{k-1}$ and set $i = 0$.
2. Calculate $P(\boldsymbol{y}|\boldsymbol{v}_i)$ for the center of the template corresponding to a grid of positions within the relevant region of interest.

3. Calculate $P(\boldsymbol{v}_i|\boldsymbol{y}) = P(\boldsymbol{v}_i)P(\boldsymbol{y}|\boldsymbol{v}_i)$
4. $i = i + 1$. Go to 2 if $i < k$.
5. Extract the initial configurations from the calculated $P(\boldsymbol{v}|\boldsymbol{y})$.

Note that the search configurations includes rotation, scaling and shape changes but not translation. We refer to *[70]* for a discussion on how to create the relevant search configurations, select the relevant region of interest and extract the initial configurations from $P(\boldsymbol{v}|\boldsymbol{y})$.

Cootes et al. [61] and Stegmann et al. *[165]* use an optimization based variant of the search strategy, which can be summarized as:

1. Create relevant search configurations $\boldsymbol{v}_0, ..., \boldsymbol{v}_{k-1}$ and set $i = 0$.
2. Perform limited MAP estimation, $\hat{\boldsymbol{v}}_i = \max_{\boldsymbol{v}_i} P(\boldsymbol{v}_i|\boldsymbol{y})$, for the center of the template corresponding to a grid of positions within the relevant region of interest. The limited MAP estimation is defined as using less than q iterations in the iterative optimization scheme.
3. $i = i + 1$. Go to 2 if $i < k$.
4. Extract the initial configurations from the calculated $P(\hat{\boldsymbol{v}}|\boldsymbol{y})$.

Note that the static search is a special case ($q = 0$) of the optimization based search.

4.2.2 Very Fast Static Search.

The disadvantages of the search strategy is the computational cost, which mainly originates from the evaluations of $P(\boldsymbol{v}|\boldsymbol{y})$. In the case of a static search Fisker et al. *[68]* propose a very fast search algorithm based on a filter interpretation of the likelihood. The algorithm exploits the fact that many likelihood models can be interpreted as a simple correlation of a filter, $f_l(\boldsymbol{v}, \boldsymbol{\theta})$, and the image, $\boldsymbol{y}$. Recall that in the static search a static template, $\boldsymbol{v}_i$, is shifted around the image and $P(\boldsymbol{y}|\boldsymbol{v})$ is calculated for each position in the grid - usually corresponding to each pixel. Using the filter interpretation all these calculations can be performed by one convolution of the filter $f_l(\boldsymbol{v}_i, \boldsymbol{\theta})$ and $\boldsymbol{y}$. Performing the convolution in the Fourier space reduces the computation time further and makes the computation time invariant to the size of the filter/template *[70]*. Fisker et al. *[68, 70]* report significant reduction in computation time using this approach. We refer to *[68, 70]* for details on the fast search strategy and experimental results.

4.3 Multi-hypothesis Initialization.

For hard initialization problems false configurations might be extracted in the initialization (or a number of true configurations are missed). Examples of hard initialization problems are many objects in the same image with large changes in scale and orientation, overlapping objects and varying background.

The simple solution is to perform a denser search. However this can be very computational expensive. An alternative strategy is to work with multiple object hypotheses, i.e. extract all reasonable initial configurations accepting some of them are false hypothesis. These initial configurations are then feed to the optimization algorithm. Finally a validation is performed to remove the false objects. Examples of multi-hypothesis initialization can be found in [61]*[66, 70, 165]*.

4.4 Conclusion and Discussion.

In this chapter a number of initialization strategies have been reviewed. The only initialization strategy, which can be applied to any deformable template (with an explicit optimization criterion) is the search strategy. Two types of search strategies are presented: static search and optimization based search. The choice of search depends on the actual problem, model, needed search configurations etc. But as a rule of thumb the static search should be applied for quite simple initialization problems, since it is faster. Whereas the optimization based search should be applied in the harder cases.

An alternative to the search based initialization is the feature based initialization. Feature based search strategies become attractive when the computational cost becomes to high using a search strategy, e.g. for objects with large variation in scale and orientation where a huge number of search configurations are needed for a successful initialization.

Given one object in an image the global optimum of $P(\boldsymbol{y}|\boldsymbol{v})$ corresponds to this object, if the model is properly designed. In this case the search based initialization guarantees successful initialization since it is only a question of how dense the search need to be to obtain an initial configuration, which achieves successfully refinement during the optimization. In general a search based approach is theoretically more appealing since a unified model is used for initialization and optimization. Whereas the feature based approach applies one model for initialization and another model for the optimization/fitting. It is hard to see theoretical arguments for applying two different models. It could be argued that if some feature is relevant for initialization it should be relevant for the optimization as well and vice versa. Basically there is only one argument for using features based initialization and that is speed.

Chapter 5

Optimization

The final optimization is typically a medium-to-high-dimensional optimization problem. In most cases, the problem can be categorized as continuous, unconstrained optimization of a nonlinear, non-convex function. The problem is continuous because the template parameters are real, i.e. $\Omega_v \subset \mathbb{R}^n$ where n is the number of template parameters, and unconstrained because most authors do not work with hard constrains.

The following sections contain a compact survey of the most popular optimization methods used with deformable template models. Recall $\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y}) = \min_{\boldsymbol{v}} U(\boldsymbol{v},\boldsymbol{y})$. In practice the MAP estimation is always performed as $\min_{\boldsymbol{v}} U(\boldsymbol{v},\boldsymbol{y})$, hence we will only deal with this in the following.

5.1 Deterministic Optimization Methods

5.1.1 Gradient Based Methods

In the optimization literature the gradient based methods receive much attention, see e.g. [52, 71] for an introduction. The gradient based methods rest on the computation of the partial derivatives of the posterior energy, $\frac{\partial U(\boldsymbol{v},\boldsymbol{y})}{\partial \boldsymbol{v}}$. These derivatives are then more or less cleverly applied in the search for the optimum. In the simple case of steepest descent the optimization is performed by iteratively updating the parameters, $d\boldsymbol{v}$, in the direction of the gradient, i.e. $d\boldsymbol{v}_t \propto -\nabla U(\boldsymbol{v}_t,\boldsymbol{y})$. The gradient based approaches are very popular with deformable template models, see e.g. [33, 72, 101, 102, 108, 163, 189].

5.1.2 Dynamic Programming

Assume the posterior can be separated into dependent subproblems:

$$\begin{aligned} U(\boldsymbol{v},\boldsymbol{y}) &= U_0(\boldsymbol{v}_0,...,\boldsymbol{v}_l,\boldsymbol{y}) + U_1(\boldsymbol{v}_1,...,\boldsymbol{v}_{l+1},\boldsymbol{y}) + ... + \\ &\quad U_{k-l-1}(\boldsymbol{v}_{k-1-l},...,\boldsymbol{v}_{k-1},\boldsymbol{y}) \end{aligned} \tag{5.1}$$

where l is a (small) integer. Then the optimization can be performed very efficiently by dynamic programming. Dynamic programming works by solving the separated energy terms $U_i(\boldsymbol{v}_i,...,\boldsymbol{v}_{i+l},\boldsymbol{y})$ recursively given $\boldsymbol{v}_{i+1},...,\boldsymbol{v}_{i+l}$ is fixed. Dynamic programming provides the machinery for enforcing hard constrains and guarantees the optimal solution within the resolution of this interval. For a more elaborate treatment of dynamic programming see e.g. [13, 44]. In the context of deformable template model dynamic programming was introduced by Amini et al. [4]. Applying dynamic programing for optimization of deformable template models have been investigated and further developed in [28, 45, 74, 110, 113, 170].

5.1.3 Difference Decomposition

In difference decomposition proposed by Gleicher [81] and in a similar approach proposed in parallel by Cootes et al. [35, 40] the template parameters, $\boldsymbol{v}$, are updated based on the difference, D, between the image, $\boldsymbol{y}$, and the present state of the image model. This approach is only applicable for deformable template models with a full pixel model. By displacing the parameters in a training set with ground truth the relationship between displacements, $\boldsymbol{dv}$, and difference images, D, are learned. The relationship is approximated by a linear regression, $\boldsymbol{dv} = RD$, where the matrix R is obtained from the training. The main difference between the approach in [81] and in [35, 40] is how R is determined. Difference decomposition has also been applied in [100, 156].

5.1.4 Heuristics

Beside the above mentioned methods numerous heuristics have been applied. The heuristics are usually tuned to a specific model, but can be very efficient. Examples can be found in [41, 60, 118].

5.2 Stochastic Optimization Methods

5.2.1 Genetic Algorithms

Genetic algorithms [82, 93] employ mechanisms similar to the evolution in nature. Genetic algorithms maintain a population of templates. This population evolves into the optimal solution by producing new generations of templates, where the survival of the individual template is based on the corresponding posterior probability. A new generation is born by mutating the existing population in a probabilistic fashion, e.g. by random crossover. We refer to e.g. [82, 135] for a more rigorous treatment of genetic algorithms. Examples of applying genetic algorithms to deformable template models can be found in [12, 50, 91, 128].

5.2.2 Simulated Annealing

Simulated annealing [27, 112] can be compared to a physical procedure – called annealing – in which a physical substance is melted and then slowly cooled in search of a low energy configuration. The system is controlled by a decreasing temperature, T_t, and new configurations are generated by a random search method, such as the Metropolis algorithm [134] or the Gibbs sampler [75]. For high T_t a new configuration with a higher energy, $U(\boldsymbol{v}, \boldsymbol{y})$, might be accepted, whereas at low T_t only configuration corresponding to an decrease in $U(\boldsymbol{v}, \boldsymbol{y})$ is likely to be accepted. The expectation is that large early fluctuations will allow rapidly escape from shallow minima whereas the latter behaviour will act as a greedy algorithm descending into the global minimum. Popular temperature schemes are: $T_t = \kappa T_{t-1}$ [112] and $T_t = \frac{c}{log(1+t)}$ [75], where κ and c are constants. Analytic conditions have been derived for the convergence towards the global optimum for a number of Markov Random Field models, see [75, 88]. Examples of the use of simulated annealing in the context of deformable models can be found in [23, 85, 155, 167]*[68, 70]*.

5.2.3 Stochastic Diffusion

Stochastic diffusion for optimization [3, 76] is based on the stochastic differential equation:

$$\boldsymbol{dv}_t = -\nabla U(\boldsymbol{v}_t, \boldsymbol{y}) + \sqrt{2T_t}\, \boldsymbol{dw}_t \tag{5.2}$$

where $\boldsymbol{w}_t$ is the standard Brownian motion with independent identical distributed components and the temperature, T_t, controls the magnitude of the random fluctuations. (5.2) can be interpreted as a steepest descent algorithm with random fluctuation in the gradient. The annealing approach ensures large fluctuation for high temperatures whereas the latter behaviour will be a steepest descent. Given special assumptions about $U(\boldsymbol{v}, \boldsymbol{y})$ global convergence have been derived, see [76, 79]. Examples of optimization of deformable template by stochastic diffusion can be found in [5, 86].

5.3 Comparative Study

In a comparative study by Fisker et al. *[68]* the performance of three gradient based methods, Simulated Annealing and Pattern Search [94] is compared for a simple ellipse model. With respect to the final energy the study shows, that all methods perform almost equally well within a reasonable number of function evaluations, but if a large number of function evaluations is allowed simulated annealing obtains very good energies. When the speed of convergence is compared, the gradient based methods are fastest during the later phase of the optimization. When the initial configuration is further away from the final solution, the relative performance of Pattern Search and Simulated Annealing increase. Basically all these results are in very good agreement with the expected behaviour.

5.4 Conclusion and Discussion

The task of optimization is not only a question of obtaining the minimal energy configuration, but also a question of convergence speed. If the objective is to obtain the minimal energy a method with good global convergence properties should be selected, i.e. dynamic programming, simulated annealing or stochastic diffusion. If concerned at the speed of convergence a deterministic method should be applied, since the stochastic methods in general suffer from slow convergence. Dynamic programming satisfies both objectives. However, it requests the separation of the energy into subproblems. In general the choice of optimization is a classical tradeoff between accuracy and speed, and it is also highly dependent on the actual model. Note that the computational cost and robustness of the optimization can be improved by a multi-resolution strategy [42, 102].

Another important aspect for the choice of optimization algorithm is the general quality of the initialization. If the initialization is close to the optimum a greedy algorithm becomes more attractive. In

general it is our experience that a good initialization often is more important for the performance than the actual choice of optimization algorithm. This conclusion is also supported by the empirical results in *[68]*.

Another important - but often forgotten - aspect for obtaining successful optimization is the quality of the optimization criterion. A properly designed optimization criterion should be smooth with few local maxima. The prior plays an important role in obtaining a smooth and convex optimization criterion, since most prior models consist of a quadratic term and hence are born smooth and convex. Jain et al. [102] utilize this by gradually reducing the smoothness of the optimization criterion by down weighting the influence of the prior during the optimization process.

Chapter 6

Conclusion and Discussion

Just the richness of the deformable template model literature documents the power of this approach. Examples of successful designs and applications of deformable models have also been demonstrated in this thesis.

However most models still suffer from a lack of generality and request expert knowledge. This even holds for most models, which are general in the sense that they can be applied to objects with an arbitrary shape. The main problem is the common difficulties of model parameter selection and initialization. In this thesis we argue that a search based approach for initialization seems to be the only general and the theoretical most appealing way to perform initialization. Due to considerations with computation time feature based initialization approaches might be attractive for objects with large Euclidean variations. However experimental results for four different models indicate that the search based approach is feasible for most problems - especially if the proposed fast filter based search is applied.

We also devise a general method for estimation of the model parameters based on a combination of a maximum likelihood and minimum distance criterion. Usually this method also give the opportunity to

simulate the prior model, which can be a powerful tool for validation. Successful model parameter estimation and the power of prior model simulations have been demonstrated for two different models.

This thesis also explores the last of the common difficulties, which is optimization. A comparative study of five different general optimization algorithms is reported. The conclusion of the study is that gradient based methods are recommended for good initialization whereas stochastic methods becomes more attractive for bad initialization or if the computation time is less important. This conclusion correspond very well with the nature of the methods. Note that dynamic programming has very good properties with respect to both computation time and global convergence. However dynamic programming is non-general, since it requires a certain separation of the optimization criterion. Even though the choice of the optimization algorithm can make a difference, it is our experience that a good initialization and a properly designed optimization criteria with few local minima often are more important factors for the overall performance.

The trends within the field of deformable template models are quite clear. The new 2D models are incorporating texture and many new models are designed for 3D (spatial and temporal). The proposed models are getting more general, but many models are still invented for applications, where one of the powerful general models would have been a better choice. Despite the popularity of deformable template models in the academic community few commercial applications seems to be based on this approach. One reason for this seems to be the requirement for expert knowledge to apply them to a new problem.

But we hope that this thesis will accelerate the use of deformable template models in particular in commercial applications.

Appendix A

Estimation of Nanoparticle Size Distributions by Image Analysis.

R. Fisker[a], J. M. Carstensen[a], M. F. Hansen[b],
F. Bødker[b] and S. Mørup[b]
[a]Dep. of Mathematical Modelling, Tech. University of Denmark
[b]Dep. of Physics, Tech. University of Denmark

Abstract

Knowledge of the nanoparticle size distribution is important for the interpretation of experimental results in many studies of nanoparticle properties. An automated method is needed for accurate and robust estimation of particle size distribution from nanoparticle images with thousands of particles. In this paper, we present an automated image analysis technique based on a deformable ellipse model that can perform this task. Results of using this technique are shown for both nearly spherical particles and more irregularly shaped particles. The technique proves to be a very useful tool for nanoparticle research.

Keywords: nanoparticles, size distribution, image analysis, deformable templates, ellipse model

A.1 Introduction.

Nanometer-sized particles currently attract considerable attention because they often have properties, which differ from those of the corresponding bulk materials [87]. Magnetic nanoparticles have important applications in, for example, magnetic recording media, ferrofluids, catalysts, and in biotechnology [87, 54]. It is characteristic for small magnetic particles that they can be superparamagnetic at finite temperatures, i.e. the magnetisation direction may fluctuate spontaneously. The dependence of the superparamagnetic relaxation time on particle size, temperature, inter-particle interactions etc. has been studied extensively during the last decades [55, 90, 115]. For the interpretation of experimental results, it is often necessary to have a detailed knowledge of the size distribution. Electron microscopy is commonly applied to estimate such particle size distributions. In order to obtain reliable size distributions it is necessary to determine the size of a large number of particles and a computer technique is therefore desirable. In this paper, we present an image analysis technique, which can be used for this purpose. We illustrate the application of the technique to electron micrographs of two different samples. One sample consists of nearly monodispersed and spherical (a-$Fe_{1-x}C_x$) particles prepared by thermal decomposition of $Fe(CO)_5$ in an organic liquid followed by oxidation in air (see Figure A.1). The second sample consists of hematite (α-Fe_2O_3) particles

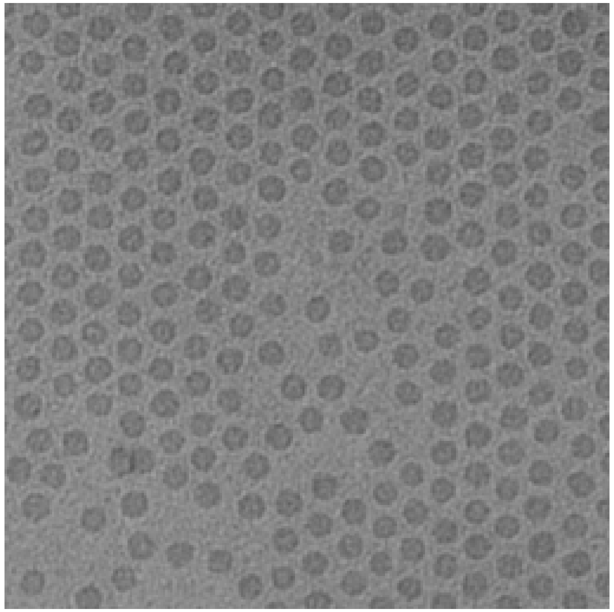


Figure A.1: Subimage of nearly monodispersed and spherical (a-$Fe_{1-x}C_x$) particles (256x256 pixels, 1 pixel ≈ 0.53 nm).

with a broader distribution in particle sizes and shapes (see Figure A.2).

A.2 The Elliptic Deformable Template Model.

Due to significant noise in the micrographs more simple segmentation approaches like noise-reducing filters followed by binarization result in highly unsatisfactory objects representing the particles. An object-oriented, as opposed to pixel-oriented, and in many cases more powerful image segmentation tool is the family of deformable template models. A deformable model can be characterized as a model, which under an implicit or explicit optimization criterion, deforms

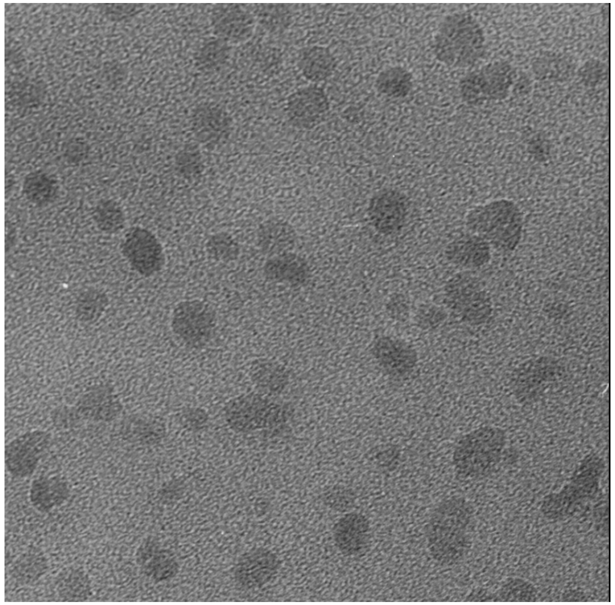


Figure A.2: Subimage of hematite (α-Fe_2O_3) particles with a broader distribution in particle sizes and shapes and a large number of overlapping particles (750x750 pixels, 1 pixel ≈ 0.20 nm).

the shape to match a known type of object in an image. For a general review we refer to [18, 101, 132]. The ideal model should be based on all pixels in the local neighborhood of each particle to suppress noise, and it should be invariant to local level changes in the image intensity, which is a very common problem. But the model should not be invariant to scale changes in the image intensity, because then all kinds of noise patterns will be picked up. These requests rule out the powerful general models [18, 33, 34, 35, 41, 85, 102, 108, 163], so a specific model needs to be designed.

The first step in the design of a new model is to choose the representation of the individual nanoparticle. In general the representation should be kept as simple as possible, i.e. the number of parameters, which describe the nanoparticle, should be low. If a high dimensional representation is chosen the model tends to overfit to noise and the optimization will become much harder and potentially unstable. Another problem in this context is overlapping particles, such as those in Figure A.2, where what seems to be one particle with a diffuse shape, often is a number of particles, which overlap. The key to separate overlapping particles is to introduce constraints on the possible shapes. One of the most efficient and popular methods is to choose a representation, which only allows certain shapes.

Based on this discussion and empirical shape observations, we choose to represent the individual nanoparticles as ellipses. In practice a nanoparticle, represented by an ellipse, is defined by the 5 parameters:

$$\boldsymbol{v} = (r_c, c_c, a, b, \phi) \tag{A.1}$$

where (r_c, c_c) is the coordinates of the center, a and b are the semi axes and ϕ is the rotation (see Figure A.3).

The next step is to design the actual match criterion, i.e. a criterion which indicates how well a given configuration of an ellipse, represented by $\boldsymbol{v}$, actually matches a nanoparticle in an image, $\boldsymbol{y}$. In a probabilistic setting this type of criterion is known as the likelihood model, $P(\boldsymbol{y}|\boldsymbol{v})$ (see e.g. [150]). Based on the required invariance to

level changes in the image intensity and the need for incorporating all pixels in the local neighborhood to suppress noise, the following likelihood model is proposed:

$$P(\boldsymbol{y}|\boldsymbol{v}) = \tfrac{1}{z_l}\exp\{-(\mu_{in}(\boldsymbol{y},\boldsymbol{v}) - \mu_{out}(\boldsymbol{y},\boldsymbol{v}))\} \tag{A.2}$$

where $\mu_{in}(\boldsymbol{y},\boldsymbol{v})$ is the mean of the pixel inside the ellipse, $\mu_{out}(\boldsymbol{y},\boldsymbol{v})$ is the mean of the pixels in a local band around the ellipse (see Figure A.3) and z_l is the normalization constant, which ensures a proper statistical distribution, i.e. $\int_{\Omega_y} P(\boldsymbol{y}|\boldsymbol{v}) = 1$, where Ω_y is the parameter space of $\boldsymbol{y}$. Based on the proposed likelihood, $P(\boldsymbol{y}|\boldsymbol{v})$, the best segmentation of a particle is obtained, when the configuration $\boldsymbol{v}$ locally maximizes the difference between the mean of the pixels inside the ellipse and the mean of the pixels in the band around the ellipse. In the ideal case this will place the ellipse on the edge between the particle and the background.

The means in (2) are defined by:

$$\begin{aligned} \mu_{in}(\boldsymbol{y},\boldsymbol{v}) &= \tfrac{1}{|\Omega_{in}(\boldsymbol{v})|}\textstyle\sum_{(r,c)\in\Omega_{in}(\boldsymbol{v})} y(r,c) \\ \mu_{out}(\boldsymbol{y},\boldsymbol{v}) &= \tfrac{1}{|\Omega_{out}(\boldsymbol{v})|}\textstyle\sum_{(r,c)\in\Omega_{out}(\boldsymbol{v})} y(r,c) \end{aligned} \tag{A.3}$$

where $y(r,c)$ is the image intensity at row r and column c, $|\Omega|$ is the number of pixels in Ω, $\Omega_{in}(\boldsymbol{v})$ and $\Omega_{out}(\boldsymbol{v})$ are the set of pixels inside the ellipse and the set of pixels in the band around the the ellipse, respectively (see Figure A.3). The sets are given by:

$$\begin{aligned} \Omega_{in}(\boldsymbol{v}) &= \{(r,c)|\tfrac{r_r^2}{a^2} + \tfrac{c_r^2}{b^2} \leq 1\} \\ \Omega_{out}(\boldsymbol{v}) &= \{(r,c)|1 < \tfrac{r_r^2}{a^2} + \tfrac{c_r^2}{b^2} \leq \rho\} \end{aligned} \tag{A.4}$$

where $r_r = (r - r_c)\cos\phi + (c - c_c)\sin\phi$, $c_r = (c - c_c)\cos\phi - (r - r_c)\sin\phi$ and $\rho - 1$ corresponds to the ratio between the area of the band and the ellipse. The choice of the relative size of the band, defined by ρ, depends on the distance between the particles and the general amount of noise.

In addition to the likelihood criterion it is very common to incorporate a prior model, $P(\boldsymbol{v})$, into the optimization criterion. A prior

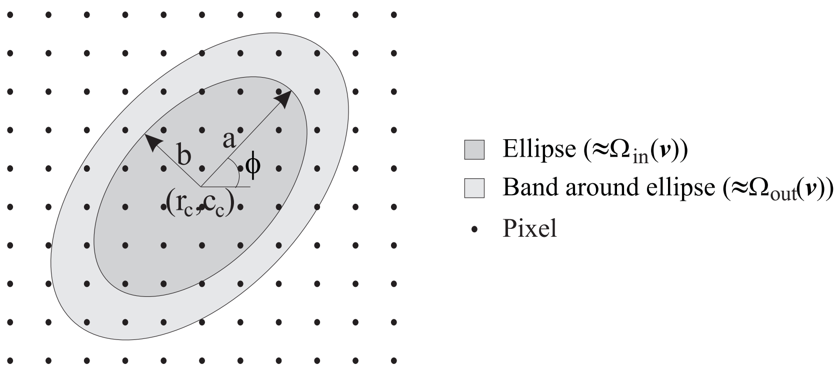


Figure A.3: Ellipse model.

model is a very powerful tool to suppress noise and stabilize the optimization, but the final segmentation result will be biased, because the full optimization criterion will favor shapes, which correspond well with the prior knowledge. For the segmentation of the nanoparticles it does not seem necessary to apply a more specific prior, so a uniform prior distribution is proposed.

The final optimization criterion, known as the posterior distribution, $P(\boldsymbol{v}|\boldsymbol{y})$, is created using Bayes theorem, $P(\boldsymbol{v}|\boldsymbol{y}) \propto P(\boldsymbol{v})P(\boldsymbol{y}|\boldsymbol{v})$ (see e.g. [150]):

$$P(\boldsymbol{v}|\boldsymbol{y}) = \tfrac{1}{z_l}\exp\{-(\mu_{in}(\boldsymbol{y},\boldsymbol{v}) - \mu_{out}(\boldsymbol{y},\boldsymbol{v}))\} \tag{A.5}$$

Because z_l is a constant, maximizing $P(\boldsymbol{v}|\boldsymbol{y}) = \frac{1}{z_l}\exp\{-U(\boldsymbol{v}|\boldsymbol{y})\}$ is identical to minimizing the posterior energy, $U(\boldsymbol{v}|\boldsymbol{y})$:

$$U(\boldsymbol{v}|\boldsymbol{y}) = \mu_{in}(\boldsymbol{y},\boldsymbol{v}) - \mu_{out}(\boldsymbol{y},\boldsymbol{v}) \tag{A.6}$$

A.3 Initialization and Optimization.

To apply the proposed model to segment the individual nanoparticles it is necessary to estimate the configurations of $\boldsymbol{v}$, which correspond

to the local minima of $U(\boldsymbol{v}|\boldsymbol{y})$. This estimation is performed using an iterative optimization procedure, which presumes an initial configuration of the parameters. This separates the estimation into two steps: initialization and optimization.

To perform the initialization, a search strategy [61, 69, 68, 101] is applied. The concept of the search strategy is to perform a sparse search in the parameter space of $\boldsymbol{v}$. In practice the search is carried out by shifting different configurations of the ellipse around the image and calculating the posterior energy $U(\boldsymbol{v}|\boldsymbol{y})$ at each position. These configurations are defined as the search configurations. The initial configurations are then extracted from the calculated energies. The full initialization algorithm can be summarized as:

1. Create relevant search configurations $(a_1, b_1, \phi_1), ..., (a_k, b_k, \phi_k)$ and choose $i = 1$.
2. Calculate $U(\boldsymbol{v}_i|\boldsymbol{y})$ for the center of the ellipse (r, c) corresponding to each pixel in $\boldsymbol{y}$, where $\boldsymbol{v_i} = (r, c, a_i, b_i, \phi_i)$.
3. $i = i + 1$. Go to 2 if $i \leq k$.
4. Extract the initial configurations from the calculated values of $U(\boldsymbol{v}|\boldsymbol{y})$.

The actual choice and number of search configurations $(a_1, b_1, \phi_1), ...,$ (a_k, b_k, ϕ_k) are determined by the amount of variation in scale, orientation and aspect ratio of the actual particles combined with the overall demand for precise initialization. For many problems it is sufficient to do the search with configurations corresponding to circles of different sizes, but for harder problems configurations with different aspect ratios need to be included.

The final step is to extract the initial configurations from the calculated $U(\boldsymbol{v}|\boldsymbol{y})$. Based on the assumption that only one template has the center in (r, c), the minimum posterior energies $U(\boldsymbol{v}|\boldsymbol{y})_{(r,c)} =$

$\min(U(r,\ c, a_i, b_i, \phi_i|\boldsymbol{y}))$ for all (r, c) are obtained. The initial configurations are then extracted as the configurations, which correspond to local minima in $U(\boldsymbol{v}|\boldsymbol{y})_{(r,c)}$ with a posterior energy below a threshold, t_{init}. A local minimum is defined as a minimum, which has the lowest posterior energy in a $\omega \times \omega$ window centered around the minimum. i.e. a local minimum in (r, c) should fulfill $U(\boldsymbol{v}|\boldsymbol{y})_{(r,c)} = \min(U(\boldsymbol{v}|\boldsymbol{y})_{(r+j,c+l)})$ for $j = -\omega/2, ..., \omega/2$ and $l = -\omega/2, ..., \omega/2$.

A disadvantage of the search strategy is the high computational cost, but the computation time is significantly reduced by a linear filter interpretation of the likelihood function [68].

The following optimization can be categorized as continuous unconstrained optimization of a nonlinear function. Fisker et al. [68] report a comparative study of the performance of 5 general optimization algorithms on a similar model. The algorithms are the gradient based steepest descent, conjugate gradient and the Quasi-Newton method BFGS and the non-gradient based algorithms pattern search [94] and simulated annealing with random walk [27, 112]. All the gradient based methods are well known optimization methods, see e.g. [52]. For initializations close to the final result all algorithms obtain the same average probability, but the gradient based method used less computation time, so BFGS is recommended. For poor initializations simulated annealing gives a statistically significant higher probability than the other methods, and in these cases simulated annealing is recommended. We refer to [68] for the detailed parameter settings for BFGS and simulated annealing.

A.4 Validation.

In the simple case where the particles do not overlap, the proposed algorithm performs very well independent on size variations, but for the case with large overlaps and size variations a number of false configurations are extracted in the initialization (or a large number of true configurations are missed). To solve this problem a final validation must be performed on the optimized result.

Essentially, false candidates only occur, when there is a high degree of overlap, so validation is only needed for these particles. The overlap is separated into two types. The first type is, when one particle is inside a single other particle. The second type is, when a number of particles overlap, i.e. the total overlap is above $t_{overlap}$ percent. A particle is defined to be inside another particle if more than t_{inside} percent of the particle is overlapping the other. When one or more small particles are inside a large particle, the fundamental question is, whether the large particle or the group of small particles is correct. The first request for the small particles is that they cover approximately the same area as the large one, otherwise the small particles do not seem to be a reasonable alternative. In practice the small particles are requested to overlap the large particle by at least $t_{overlap}$ percent. In the case with a overlap larger than $t_{overlap}$, the energy of the large particle and the average energy of the group of small particles is compared, and the one with the highest energy is removed.

This still leaves the second type of overlap, where groups of particles overlap more than the maximal value, $t_{overlap}$. To correct this, the particle with the largest overlap is removed in each group until all particles are below $t_{overlap}$.

A.5 Experimental Results.

The proposed model and algorithm have been applied on two qualitatively different samples. The first sample consisted of ferromagnetic particles of the amorphous alloy a-$\mathrm{Fe}_{1-x}\mathrm{C}_x$ (x ≈ 0.22) prepared by thermal decomposition of iron penta-carbonyl ($\mathrm{Fe(CO)}_5$, 20.0 cm^3) in an organic liquid (cis-trans decalin, 50.0 cm^3) in the presence of surfactant molecules (oleic acid, 4.0 g) by the method described in [178]. The size and distribution of sizes of the resulting particles depend on the choice and amount of surfactant and the heating conditions. The preparation technique leads to the formation of well-separated, surfactant coated particles with nearly spherical shape and with a very narrow size distribution (see Figure A.1). This fact makes these particles interesting for basic studies of the dynamic magnetic behavior of ensembles of nanoparticles. The magnetic properties of magnetic dispersions with different volume concentrations of the particles have been studied using ac and dc magnetometry. The main focus of these studies has been to compare the dynamic behavior with that of interacting atomic dipoles and in particular with that of spin-glass materials [53, 89, 106]. The particles oxidize in contact with air. Therefore, the TEM studies were performed on grids with oxidized particles prepared by placing a droplet of the ferrofluid on the grid and leaving the grid in air until the particles had fully oxidized.

The nearly spherical shapes of the nanoparticles in this sample correspond very well with the ellipse representation. The first step in the segmentation of the particles is to perform the initialization. Based on the distance between the particles, the area of the band is chosen to be identical to the area of the ellipse, i.e $\rho = 2$. Due to the limited variation in shape and size, only two search configurations are used. The configurations are two circles with radius 5 and 6 pixels, i.e. $a_1 = b_1 = 5, a_2 = b_2 = 6$ and $\phi_1 = \phi_2 = 0$. The corresponding minimum posterior energy image $U(\boldsymbol{v}|\boldsymbol{y})_{(r,c)}$ for the subimage in Figure A.1 is shown in Figure A.4. The initial configurations are then extracted as the configurations, which correspond to local minima in the posterior energy image. The window size ω is selected to the smallest radius in the initial configurations, i.e $\omega = 5$, so the distance between the center of two initial configurations is at least 3 pixels. The next step is to select the threshold, t_{init}, which determines the sensitivity of the method with respect to separating particles and background. Based on empirical experience t_{init} is selected to $t_{init} = -10.5$, which according to the definition of $U(\boldsymbol{v}, \boldsymbol{y})$ corresponds to a difference of 10.5 between the average intensities inside the ellipse and in the band around the ellipse. The initial configurations are extracted and optimized using BFGS (see Figure A.5). Due to the limited number of overlapping particles no automatic validation is performed. The segmentation is manually validated (see Table A.1), which is quite easy for these particles. As many as 99.8% of the particles are segmented correctly and all the errors (missed and incorrect) correspond to overlapping particles. Another important result is that as many as 4545 particles are segmented. Such a number of measurements would have been extremely time consuming to obtain manually. The total computation time for the segmentation of the full image on a PC, Pentium II 350 Mhz, was 15.2 min. (initialization: 0.2 min., optimization: 15.0 min.). In general the computation time is highly dependent on a number of factors, where some of the most important are the number and size of the particles.

Particles	a-$Fe_{1-x}C_x$	α-Fe_2O_3
Correct	4545 (99.8%)	575 (94.3%)
Incorrect	4 (0.1%)	22 (3.6%)
Missed	7 (0.1%)	13 (2.1%)

Table A.1: Manual evaluation of segmentation. The table shows the number of correctly segmented, incorrectly segmented and missed nanoparticles.

From the segmented particles with the incorrectly segmented particles removed, the distribution of the approximated diameters, $d \equiv 2\sqrt{ab}$, the approximated volumes, $V \equiv \frac{4}{3}\pi(ab)^{3/2}$, and the aspect ratios of the particles, $\equiv \max(a, b)/\min(a, b)$, are calculated (see Figures A.6, A.7 and A.8). Gaussian and lognormal distributions are fitted to the diameter and the volume distributions (see Figure A.6-A.7 and Table A.2). To measure the goodness of fit, [153] recommend to use the Kolmogorov statistic, D, for a sample size greater than 2000 and Shapiro-Wilk statistic, W, for a sample size less than 2000. Beside the test statistic the corresponding p-value is calcu-

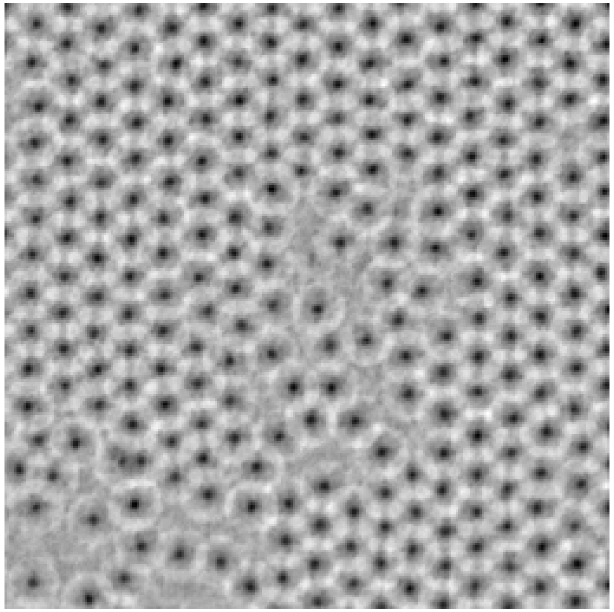


Figure A.4: Minimum posterior energy $U(\boldsymbol{v}|\boldsymbol{y})_{(r,c)}$ used for initialization of the a-Fe$_{1-x}$C$_x$ nanoparticles. The dark spots correspond to low energy, i.e. nanoparticles.

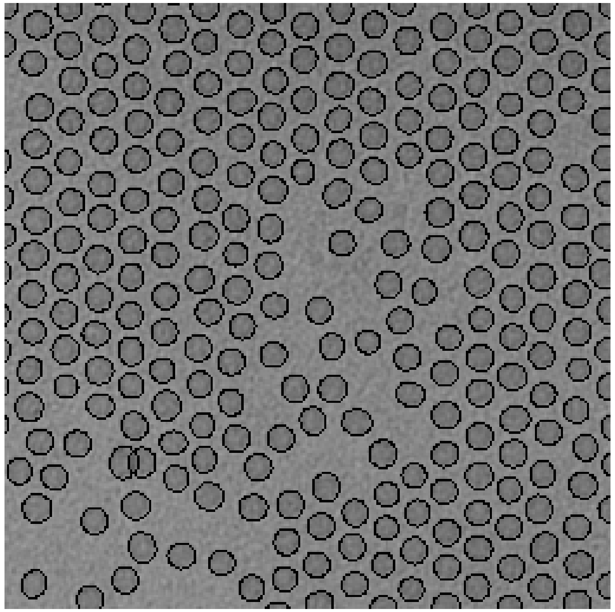


Figure A.5: Final segmented a-Fe$_{1-x}$C$_x$ nanoparticles.

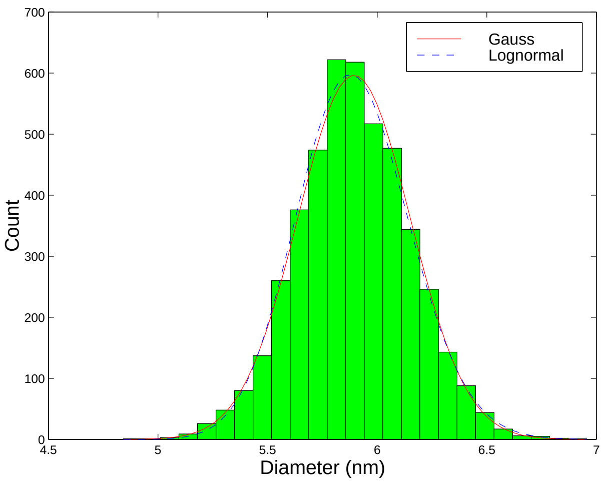


Figure A.6: Distribution of approximated diameters of a-$Fe_{1-x}C_x$ nanoparticles.

lated. The interpretation of the p-value is, that if the fitted model were correct, discrepancies as large or larger than the one observed would take place p% of the time [150], i.e. a large p-value, which corresponds to a small D, indicates a good fit. Based on the test statistics the best model for the diameter distribution is the Gaussian and for volume distribution it is the lognormal (see Table A.2). Visually the Gaussian and the lognormal distributions seem to describe the empirical distributions very well, but based on the p-values the hypothesis that the observations originate from the fitted distribution is rejected for all the fits, except the Gaussian fit to the diameter distribution. However, testing for goodness of fit can be misleading. For sufficiently good statistical quality of the data, the goodness of fit criteria will reject any parametric distribution because it is only an approximation to the empirical distribution [150].

The second sample consists of hematite nanoparticles produced according to the gel-sol method developed by [169]. First a sample of

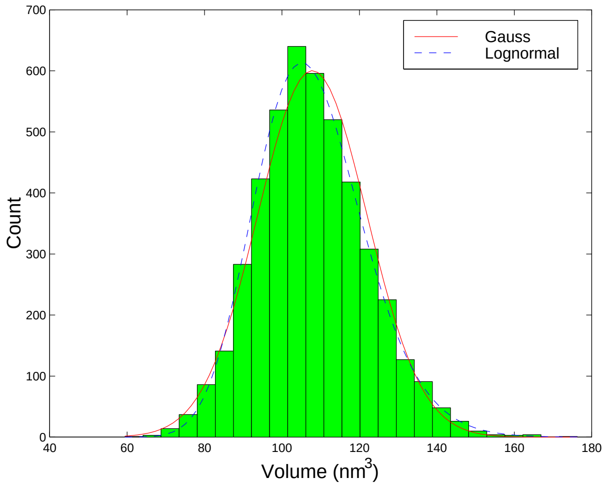


Figure A.7: Distribution of approximated volumes of a-$Fe_{1-x}C_x$ nanoparticles.

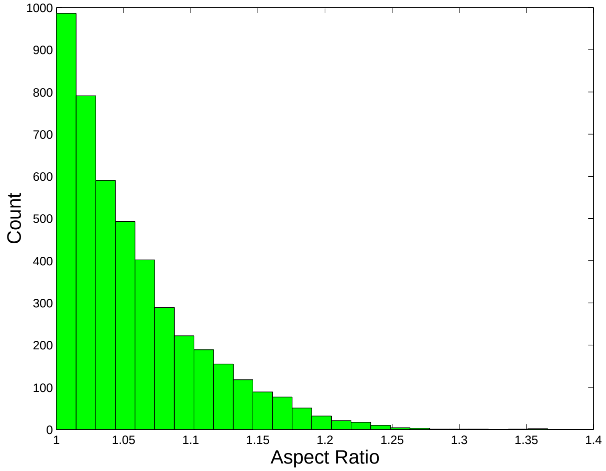


Figure A.8: Distribution of aspect ratios of a-$Fe_{1-x}C_x$ nanoparticles.

Observations	Distribution	Estimated Parameters	Goodness of fit
Diameter	Gauss	$\mu_d = 5.89$ nm, $\sigma_d = 0.26$ nm	$D = 0.013, p = 0.05$
a-$Fe_{1-x}C_x$	Lognormal	$\alpha_d = 5.9$ nm, $\beta_d = 0.044$	$D = 0.016, p \leq 0.01$
Volume	Gauss	$\mu_V = 108$ nm^3, $\sigma_V = 14$ nm^3	$D = 0.031, p \leq 0.01$
a-$Fe_{1-x}C_x$	Lognormal	$\alpha_V = 107$ nm^3, $\beta_V = 0.13$	$D = 0.016, p \leq 0.01$
Diameter	Gauss	$\mu_d = 8.12$ nm, $\sigma_d = 2.1$ nm	$W = 0.967, p < 0.01$
α-Fe_2O_3	Lognormal	$\alpha_d = 7.87$ nm, $\beta_d = 0.25$	$W = 0.966, p < 0.01$
Volume	Gauss	$\mu_V = 337$ nm^3, $\sigma_V = 260$ nm^3	$W = 0.878, p < 0.01$
α-Fe_2O_3	Lognormal	$\alpha_V = 270$ nm^3, $\beta_V = 0.66$	$W = 0.966, p < 0.01$

Table A.2: Properties of fitted distributions (number-weighted). The third column contains the estimated parameters for the Gaussian density ($f(x)dx = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})dx$) and for the lognormal density ($f(x)dx = \frac{1}{x\beta\sqrt{2\pi}}\exp(-\frac{\ln^2(x/\alpha)}{2\beta^2})dx$). The fourth column contains the Kolmogorov statistic, D, or the Shapiro-Wilk statistic, W, and the corresponding p-values.

relatively large polycrystalline hematite particles was prepared by the gel-sol method. Some of the subcrystallites were then separated from the larger polycrystalline particles by low energy ball milling the powder in an agate container with agate balls. The sample was dispersed in diluted HNO_3 and the remaining agglomerates were removed by centrifugation, resulting in an ionic suspension [8] of a small amount of hematite nanoparticles. Using this particle suspension as seeds in a second gel-sol preparation, a powder of particles with a typical size of about 60 nm was obtained. These particles are polycrystalline and the individual subcrystallites can be separated relatively efficiently by a treatment with ball milling and centrifugation as before. The final ionic dispersion of hematite nanoparticles is relatively stable and can be diluted with water and added to a grid for TEM studies.

Due to the broader distribution in shape and size and the large number of overlapping particles, the analysis of the electron micrographs of the second sample is more difficult than that of the first sample. Although the particle shapes are not strictly elliptic, the elliptic shape still seems to be a reasonable low dimensional representation. Circles are still used as initial configurations, but because of the variations in size, a much larger number of initial configurations is needed. In practice $a_i = b_i = 13 + i$ and $\phi_i = 0$ for $i = 1, ..17$ are used. For this problem $\rho = 2$, $\omega = 13$ and $t_{init} = -9$ are applied. The initialization of the particles in Figure A.2 is shown in Figure A.9. Due to the shape variation, the initial configurations are further away from the minimum energy configuration than the first sample, so simulated annealing is used for the optimization (see Figure A.10). Due to larger particles a higher variance of the random walk and 1000 iterations are used. The used variance is $\sigma_{r_c}^2 = \sigma_{c_c}^2 = 2, \sigma_a^2 = \sigma_b^2 = 1$ and $\sigma_\theta = 0.2$ (we refer to [68] for a detailed description of the meaning of the symbols). The last step is then to perform the validation. If the particles overlap by more than 70% with one or more other particles, it is very hard to distinguish between each individual particle. On this basis the validation

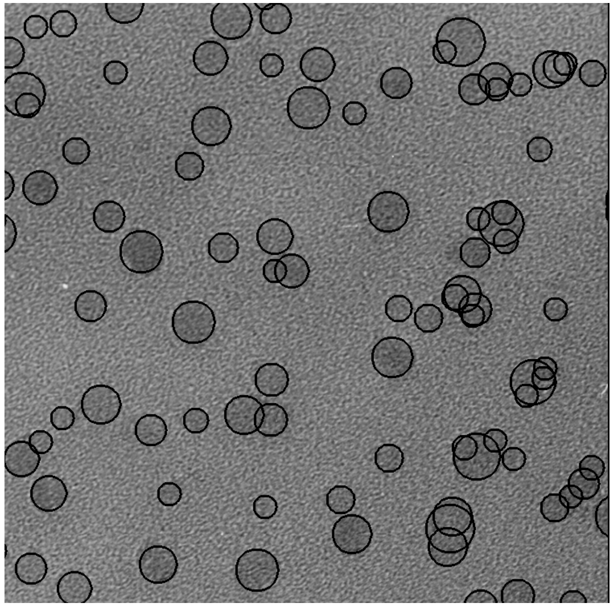


Figure A.9: Initialized α-Fe_2O_3 nanoparticles. Note that a number of false ellipses are extracted as initial configurations.

parameters are selected to $t_{overlap} = t_{inside} = 70\%$. The result of the validation is shown in Figure A.11.

The manual evaluation of the segmentation is summarized in Table A.1. A nanoparticle is evaluated as correctly segmented, if the shape and size correspond reasonable well with the "true particle". It should be stressed that this evaluation is subjective, and the manual evaluation only indicates the performance. The incorrectly segmented particles can be separated into 4 types: large carbon particle (6), border problems (5) and poor segmentation (11). The large carbon particle is a foreign object, which ideally should not be in the image. The incorrectly segmented particles along the border could be removed by erasing all segmented particles within a certain dis-

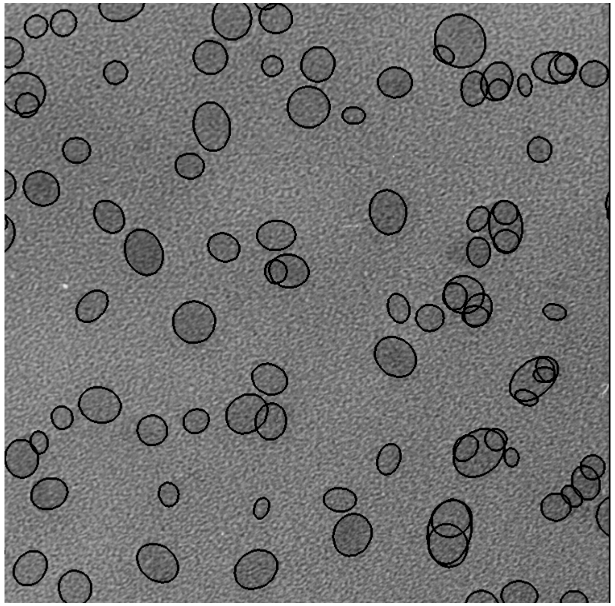


Figure A.10: Optimized α-Fe_2O_3 nanoparticles.

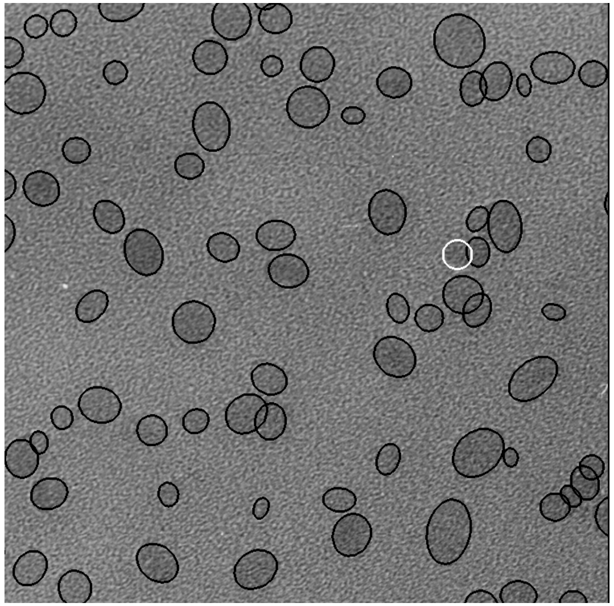


Figure A.11: Final segmentation of α-Fe_2O_3 nanoparticles after validation. The white ellipse indicates a missed particle.

tance from the border at the expense of an increasing number of missed particles. The poorly segmented particles indicate more serious problems, where the optimization is caught in a local minimum, which does not correspond to the "true particle". But this is still a quite small number compared to the overall number of particles. Most of the missed particles correspond to particles, which overlap with other particles (see Figure A.11). The total computation time for this segmentation on a PC, Pentium II 350 Mhz, was 210.2 min. (initialization: 79.5 min., optimization: 122.6 min., validation: 8.1 min.).

The distributions of the approximated diameters, the approximated volumes and the aspect ratios of the particles are shown in Figures A.12, A.13 and A.14. The distribution of the approximated diameters and volumes are again fitted by a Gaussian and a lognormal distribution function. Based on the Shapiro-Wilk statistic, W, the Gaussian and the lognormal describes the diameter distribution equally well, whereas the lognormal distribution obtain the best fit to the volume (see tabel A.2). Note that large values of the Shapiro-Wilk statistic, W, indicates a good fit. If tests for goodness of fit are performed, the tests reject that the observations originate from the fitted distributions.

A number of tests has been performed to test the sensitivity of the parameters, which need to be selected. In general the algorithm is quite insensitive to reasonable values of the parameters. In most cases the main difference is the total computation time. But it should be stressed that t_{init} is far more important than the other parameters, which can be chosen by simple rules of thumb. t_{init} is intuitively easy to choose, because it corresponds to the minimum distance between the average intensity inside and outside the particle.

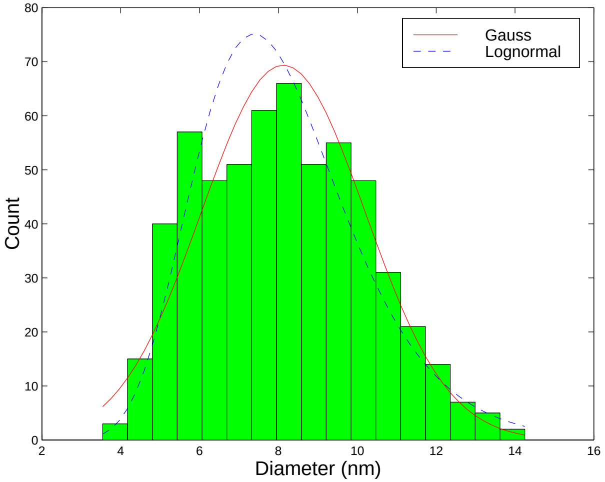


Figure A.12: Distribution of approximated diameters of α-Fe_2O_3 nanoparticles.

A.6 Discussion.

The image analysis technique, discussed in this paper, appears to be very useful for analysis of the particle size distribution in electron micrographs with large numbers of particles. In particular it gives reliable estimates of the size distribution when the particles are well separated and the particle shape can be well approximated with ellipses. This is the case for the oxidized a-$Fe_{1-x}C_x$ particles (Figure A.1). An electron micrograph of the same sample has also been analysed manually using a magnifier with a built-in scale. In the analysis, a spherical particle shape was assumed and 254 particles were analysed. The resulting size distribution of oxidized particles was well described as a lognormal distribution with the geometric mean $\alpha_d = 5.79$ nm and the logarithmic standard deviation $\beta_d = 0.07$ (This corresponds to a diameter of about 4.6 nm before oxidation). The estimated parameters for a Gaussian distribution are $\mu_d = 5.83$

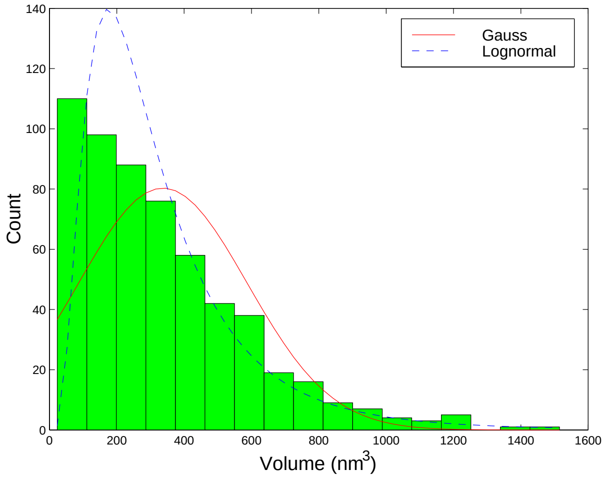


Figure A.13: Distribution of approximated volumes of α-Fe_2O_3 nanoparticles.

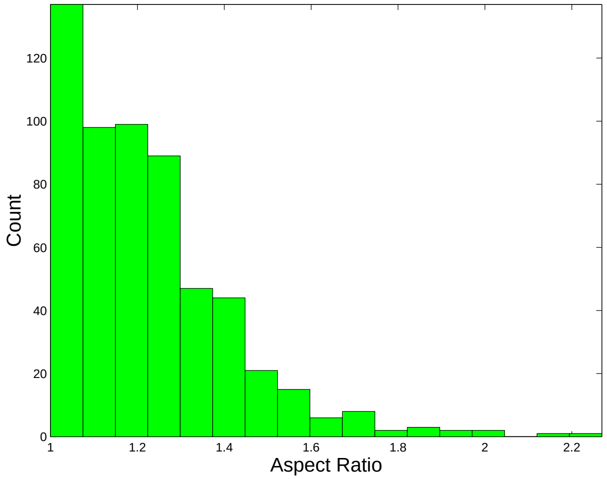


Figure A.14: Distribution of aspect ratios of α-Fe_2O_3 nanoparticles.

nm and standard deviation $\sigma_d = 0.42$ nm. The mean values are in good agreement with those obtained in the present work, but the standard variations are larger (see table A.2). This is presumably related to the larger uncertainty in the manual estimates.

The manually obtained volume-weighted volume distribution is well described by a lognormal distribution with geometric mean volume $\alpha_V = 103.7$ nm^3 and logarithmic standard deviation $\beta_V = 0.22$. The corresponding values obtained by the technique described in this paper are $\alpha_V = 113.5$ nm^3 and $\beta_V = 0.13$. Thus we find again that the mean values agree quite well, but the standard deviation is smaller when using image analysis. The size distribution is reflected in magnetic measurements. A fit to a zero-field cooled magnetization curve measured on a very dilute ferrofluid with very weak inter-particle interactions using a volume-weighted lognormal distribution yielded $\beta_V = 0.20$ [89]. The discrepancy may be related to the influence of surface effects on the magnetic anisotropy constant. It should also be noted that the fact that only 2D projections of the particles are observed may result in an incorrect size distribution and smaller aspect ratios than those of the real particles.

The use of the image analysis technique is more problematic in cases when there is overlap of particles and the particle shape is more irregular, like the α-Fe$_2$O$_3$ particles, shown in Figure A.2. However, any technique that could be used for analysis of the size distribution in such images would have difficulties estimating the size of overlapping particles. The results obtained by use of the analysis described in this paper are therefore at least as reliable as those estimated by use of other techniques. Furthermore, the present technique has the advantage that the distribution can be estimated empirically with good statistics.

The size distributions of nanoparticles are commonly well approximated by lognormal distribution functions [84]. It is also clear that the size distribution of the α-Fe$_2$O$_3$ particles is better approximated by a lognormal distribution than by a Gaussian distribution (Figures A.12-A.13 and Table A.2). The diameter distribution of the oxidized a-Fe$_{1-x}$C$_x$ particles is well fitted by both types of distribution functions, but the fit to the Gaussian distribution is slightly better than the fit to a lognormal distribution (see Figures A.6-A.7 and Table A.2). This observation is in accordance with the results of O'Grady and Bradbury [141], who found that nanoparticles, prepared by carbonyl decomposition, are better fitted to Gaussian distributions, while other preparation techniques commonly lead to size distributions that are better fitted to lognormal distributions. The volume distribution of the oxidized a-Fe$_{1-x}$C$_x$ particles is, however, better fitted with a lognormal distribution.

In non-spherical ferromagnetic particles, shape anisotropy can give a significant contribution to the total magnetic anisotropy. The a-Fe$_{1-x}$C$_x$ particles prepared by carbonyl decomposition are amorphous [178, 179], and therefore the magnetocrystalline anisotropy is negligible. The shape anisotropy of a particle with the average aspect ratio ($\max(a, b)/\min(a, b) \cong 1.05$) and the same magnetisation of the current particles is, however, about a factor of 10 smaller than the measured value. Therefore, other contributions (e.g. surface anisotropy) seem to be more important.

A.7 Conclusion.

In many studies of the properties of nanoparticles a detailed knowledge of the size distribution is important for the interpretation of experimental results. The technique presented in this paper seems to be very useful for this purpose. In particular it can easily handle a large number of particles and it is much faster and less subjective than commonly used manual techniques.

The technique has, of course, limitations, especially if the particles overlap and if the particle shape is not well represented by the used template. Uncertainties in the estimated volume distributions can also be expected because one observes 2D images of 3D objects. However, these limitations are equally important when other techniques (e.g. manual) are used to analyse the micrographs. We therefore can conclude that the technique, demonstrated here, is both faster and more reliable than other commonly used techniques.

It should be noted that the model and algorithm has been developed to segmentation of nanoparticles, but it can directly be applied to segmentation of other particles and objects, which are well approximated by ellipses and have a similar intensity distribution, e.g. cells.

Acknowledgements

We would like to acknowledge Torsten Dörge for the initial work on the elliptic deformable model.

Appendix B

Automated visual inspection of textile

Rune Fisker & Jens Michael Carstensen
Dep. of Mathematical Modelling, Tech. University of Denmark

Abstract

A method for automated inspection of two types of textile is presented. The goal of the inspection is to determine defects in the textile. A prototype is constructed for simulating the textile production line. At the prototype the images of the textile are acquired by a high speed line scan camera. The vertical threads are located using a vertical projection of the image. It is thereby possible to identify the defects in the vertical threads. A structural model of the horizontal threads is formulated. The model consists of a Markov random field, which represents a priori knowledge about the position and structure of the horizontal threads and an observation model that incorporates knowledge about the visual appearance

of the threads given their position and structure. Using this model the horizontal threads are located. Features are calculated from the located threads to identify the defects. To go from the prototype to a production line system we only need to gain a speed factor of 4.

Keywords: pattern recognition, image analysis, automated inspection, textile, line scan, Markov random fields

B.1 Introduction

In large parts of the textile industry the quality of the produced textile is measured by manual inspection. Manual inspection is labour intensive and is subject to the traditional problems related to objectivity and reproduction of a measurement. Besides this it is very difficult to ensure that defects get through the manual system undetected. Undetected defects are very expensive in terms of customer complaints.

The visual inspection system focus on two types of textiles shown in figure B.1. The perfect textile can be described as a homogeneous pattern where the vertical and the horizontal threads are orthogonal, all the threads are straight, and the distances between consecutive threads are constant. A given textile has to be classified as first, second, or third quality where the quality is determined by the kind and size of defects in the textile. The possible defects in the vertical threads are missing vertical threads or too large deviations in the distance between consecutive vertical threads. In the horizontal threads the possible defects are oblique threads, oblique borders of the threads, waving threads, or too large deviation in the distance between the threads. The width of the textile is 1 meter and the length is almost infinite. In conjunction with a max production speed of 25 meter/min. the system have to inspect 0.42 meter^2/second continuously .

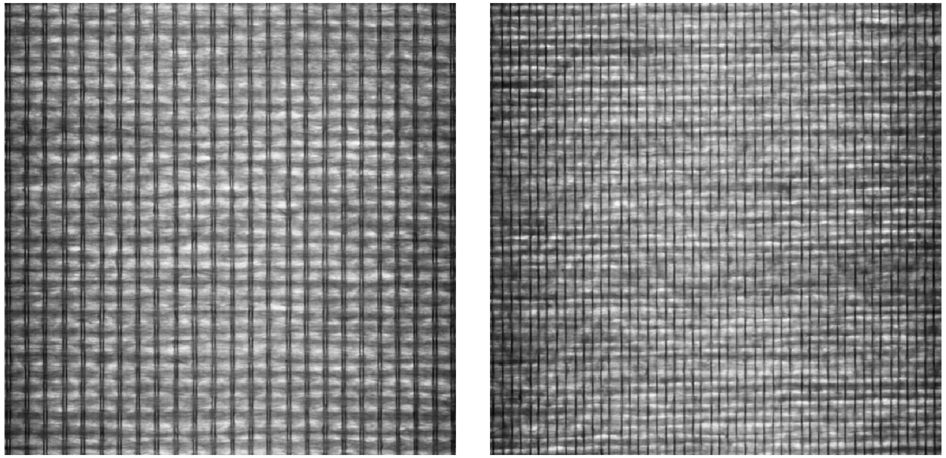


Figure B.1: Examples of the two types of textiles with ordinary weave (left) and herringbone weave (right), which have to be inspected. The images are acquired using transmission. The horizontal size of the textile is 15.75 cm and the vertical size is 17.75. With the chosen resolution the image is 630 x 630.

Automated inspection of textile quality have earlier been investigated using a Gaussian Markov Random Field [32], first-order gray level statistics [107] and second-order gray level statistics in Gaussian pyramids [24], Wavelet transform followed by a threshold [105], an enhancement filter followed by a threshold [140] and simple features feed to a Neural network [138, 172]. We presents a new method based on a vertical projection and a stochastic model of the textile pattern.

B.2 Detection of defects in the vertical threads

If the textile is moved in a constant direction the vertical threads will always be vertical in the image because of the way the textile is woven. This can be utilized to locate the position of the vertical threads by making a vertical projection of the image. The vertical projection of an image is made by calculating the average intensity for each column. In the vertical projection of an textile image the threads are located as local minima and the centre of the vertical spaces as local maxima. Given the knowledge of the position of the vertical threads the distance between the threads are directly determined. Based on the distance it is possible directly to detect deviations in the distance between the threads. From the distance it is also possible to detect missing vertical threads because they will appear as app. twice the normal distance between two threads. To determine the quality of a textile limits for the maximal deviation in the distance for each quality class are defined. An example of a vertical projection is shown in figure B.2. To suppress the noise the projection is smoothed by a Gaussian filter. In the projection are drawn the located vertical threads as vertical lines. In the figure is shown a plot of the distances. Notice the peak between thread number 12 and 13, which corresponds to the large distance between thread 12 and 13 in the image. Additionally the deviation limits

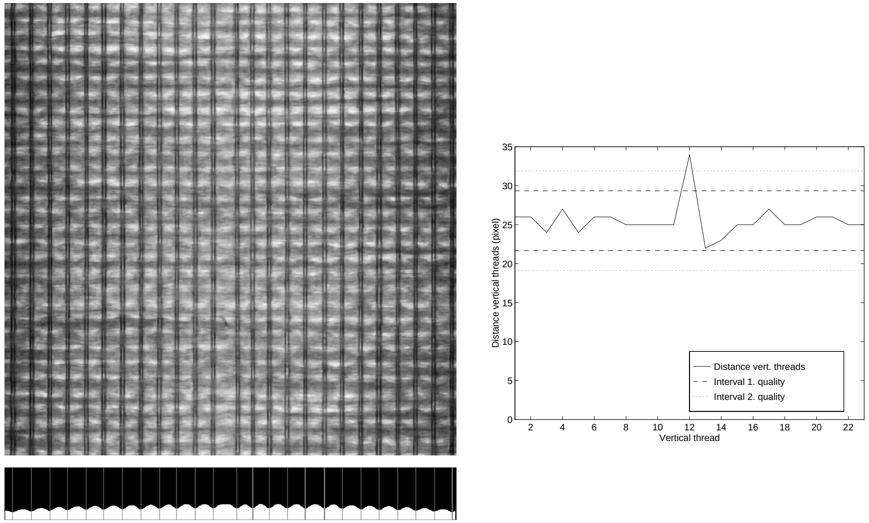


Figure B.2: Textile image (top, left). Vertical projection of image where the located vertical threads are drawn as vertical lines (bottom, left). Plot of distance based on the located threads. Notice the large peak in the plot, which corresponds to the large distance between thread 12 and 13 (right).

for first, second and third quality are drawn in the plot. Because the deviation is outside the interval for first and second quality the textile is classified as third quality.

B.3 Detection of defects in the horizontal threads

The detection of defects in the horizontal threads are also based on localization of each individual thread. In order to locate the horizontal threads a model of the threads is formulated. The continuous thread k is represented by one point in each centre of a vertical space.

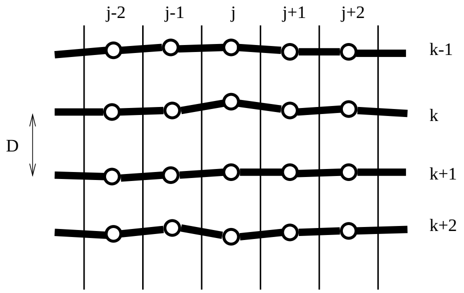


Figure B.3: Model of horizontal threads.

Thread k is thus modelled as

$$\boldsymbol{v}_k = (v_{k,0}, v_{k,1}, v_{k,2}, ..., v_{k,L-1}),$$

see figure B.3. To reduce the complexity of the model the horizontal position of the vertical space j is assumed known since it easily can be located at the same time as the vertical threads.

Based on the representation of the threads a deformable template [23, 85, 114, 154, 189] is formulated, which incorporates the information in a given image with a template of the textile structure. Given an image $\boldsymbol{y}$ and the thread $\boldsymbol{v}_{k-1}$ the a posteriori distribution of the deformable template for $\boldsymbol{v}_k$ using a Markov Random field is given as:

$$\begin{aligned} P(\boldsymbol{v}_k|\boldsymbol{y},\boldsymbol{v}_{k-1}) \quad \propto \quad & \exp\{-\alpha_0 \sum_{j=1}^{L-1}(v_{k,j}-v_{k,j-1})^2 \\ & -\alpha_1 \sum_{j=0}^{L-1}(v_{k,j}-v_{k-1,j}-D)^2 - \alpha_2 \sum_{j=0}^{L-1} M_m(v_{k,j},j)\} \end{aligned}$$

where the two first terms contains the template of the structure. The first term favours strictly horizontal threads. The second term describes that the thread $\boldsymbol{v}_k$ should be placed in a predefined distance D from $\boldsymbol{v}_{k-1}$. The third term describes the interaction of an observed image and a particular realization of $\boldsymbol{v}_k$ where $M_m(v_{k,j}, j)$ is the horizontal mean at the vertical position $v_{k,j}$ of the pixels in the vertical space j. α_0, α_1 and α_2 are model constants, which determine the relative weight of each term.

Based on the a posteriori distribution the next horizontal thread is located in a three step procedure:

1. Locate a point on the next thread.
2. Search for the rest of the next thread.
3. Validate.

To locate a point on the next thread it is assumed that a subset of the next thread $\boldsymbol{v}_{k,[m,l]} = (v_{k,m}, v_{k,m+1}, .., v_{k,l})$ locally will have the same shape as the previous thread and will be placed in the distance 0.5 to 1.5 D relatively to that thread. This assumption can be described as:

$$\boldsymbol{v}_{k,[m,l]} \quad = \quad \boldsymbol{v}_{k-1,[m,l]} + \boldsymbol{i}d, d \in [0.5D, 1.5D]$$

where $\boldsymbol{i}$ is the vector (1,1,..,1) with the same length as $\boldsymbol{v}_{k-1,[m,l]}$. A subset of the next thread $\hat{\boldsymbol{v}}_{k,[m,l]}$ is then located by determining the value $\hat{d}$, which maximizes the a posteriori distribution $P(\boldsymbol{v}_{k,[m,l]}|y, \boldsymbol{v}_{k-1})$. The point on the next thread $\hat{v}_{k,j} \in \hat{\boldsymbol{v}}_{k,[m,l]}$ is then located at the point, which have the highest a posteriori probability.

Assume that one or more points on the next thread $\boldsymbol{v}_{k,[m,l]}$ are located. The next point to the right on the thread $v_{k,l+1}$ is then located at the position $\hat{v}_{k,l+1}$, which maximizes the a posteriori distribution. Conditioning only on already located points and using that the a posteriori distribution is a Markov Random field the next point $\hat{v}_{k,l+1}$ can be located as the position, which maximizes:

$$\begin{aligned} P(v_{k,l+1} = q|\boldsymbol{y}, v_{k,l}, v_{k-1,l}) \quad \propto \quad & \exp\{-\alpha_0(q - v_{k,l})^2 \\ & -\alpha_1(q - v_{k-1,l+1} - D)^2 - \alpha_2 M_m(q, l+1)\}, \\ & q \in [v_{k,l} - C, v_{k,l} + C] \end{aligned}$$

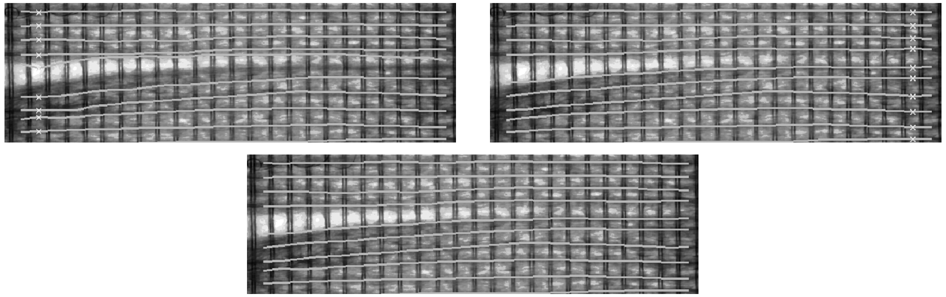


Figure B.4: The result of the search from left (top,left) and right (top,right) respectively with $\alpha_0 = 4$, $\alpha_1 = 2.5$, $\alpha_2 = 15$ and $\psi = 20\%$. The crosses marks the point found on the next thread. The result of the validation based on the leftward and rightward searches(bottom).

where C is a constant defined as the maximum shift of the thread between two vertical spaces. An analogous expression can be derived to locate points to the left.

To validate the result two searches are made one from the left side of the textile $\boldsymbol{v}_k^l$ and one from the right $\boldsymbol{v}_k^r$. These are compared point $v_{k,j}^l$ by point $v_{k,j}^r$. If the difference between the points is larger than a predefined percentage ψ of D it is assumed that there is an error in one of the searches. After detecting an error the whole interval where the difference is larger than ψ percent of D is determined. Then the a posteriori probability for each interval is determined and the interval with the highest a posteriori is assumed to be correct and is used in the final result. If there is no large deviation between the searches the thread with the highest a posteriori is used.

The three step procedure is illustrated in figure B.4 with $\alpha_0 = 4$, $\alpha_1 = 2.5$, $\alpha_2 = 15$ and $\psi = 20\%$. The two upper images are the result of the search from left and right respectively. The crosses mark the point found on the next thread. The bottom image is the result of the validation based on the searches from left and right. Notice that the final result is a combination of the two searches.

B.3.1 Defects Analysis

After the horizontal threads are located they have to be analyzed for defects. For this purpose a number of features are defined, which make it possible to detect the presence and the size of the defects. The features are:

- Mean distance $d_k = \frac{1}{L}\sum_{j=0}^{L-1} v_{k,j} - \frac{1}{L}\sum_{j=0}^{L-1} v_{k-1,j}$
- Local distance $d_{k,j} = v_{k,j} - v_{k-1,j}$.
- The gradient a_k of the thread
  Thread v_k is approximated by a linear thread $a_k h(j) - b_k$, where a_k is the gradient, b_k is the position of the linear thread at the vertical axis, and $h(j)$ is the horizontal position of vertical space.
- The linear variation δ_k of the gradient a_k defined as $\delta_k = a_k - \mu_a$ where μ_a is the mean gradient of the threads.
- 2. moment $\sigma_k^2 = \frac{1}{L}\sum_{j=0}^{L-1}(v_{k,j} - a_k\, vm(j) - b_k)^2$
- 3. moment $\gamma_k = \frac{1}{L}\sum_{j=0}^{L-1}(v_{k,j} - a_k\, vm(j) - b_k)^3$

The mean and local distances are used for detection of deviations in the distance between the horizontal threads. The gradient of the thread is used for detection of oblique threads. The 2. moment, which can also be described as the linearity of the thread is used for detection of waving threads. The linear variation of the gradient and the 3. moment is used for detection of threads with oblique border. The 3. moment can be interpreted as the skewness of the thread.

B.4 Experiments

In order to analyze images of the same quality as in the production a hardware prototype is made, which simulates the production. The

textiles are moved forward using a conveyer belt and the image acquisition is made by a high speed line scan camera. To get the best contrast in the images transmission (background light) is used. The light has to be high frequency because 50 Hz light will be superimposed in the images. All presented images are acquired using the prototype.

Images of 22 pieces of textile have been acquired at the prototype. Three images each representing 20 % of the original image are shown in figure B.5, B.6 and B.7. In the textiles the located threads are drawn. The dark threads indicates that there have been detected an error. Notice the threads are located correct. Based on the located threads the features for the defect analysis are calculated and plotted in the figures.

The textile in figure B.5 contains deviation in the distance between the horizontal threads. Note the peaks in the plot of the distance between the horizontal threads, which indicate the deviation in the distance. Notice also the small oscillations in the rest of the features as expected because there is no other defects. Figure B.6 contains a textile with oblique threads, which is clearly indicated by the high gradient in the plot of the gradient and the small oscillations in the rest of the features. The last textile in figure B.7 have an oblique border, which is revealed by the plot of the 3. moment, 2. moment, and the linear variations of the gradient. Note how the method works for both the two difference types of textiles.

B.5 Conclusion

A method which successfully inspects two types of textiles and determine the quality has been presented. The quality is determined by the presence and size of the defects which appear in the textile. The defects in the vertical threads are found by locating the vertical threads using a vertical projection of the image. The horizontal threads are located using a structural model of the threads. Based

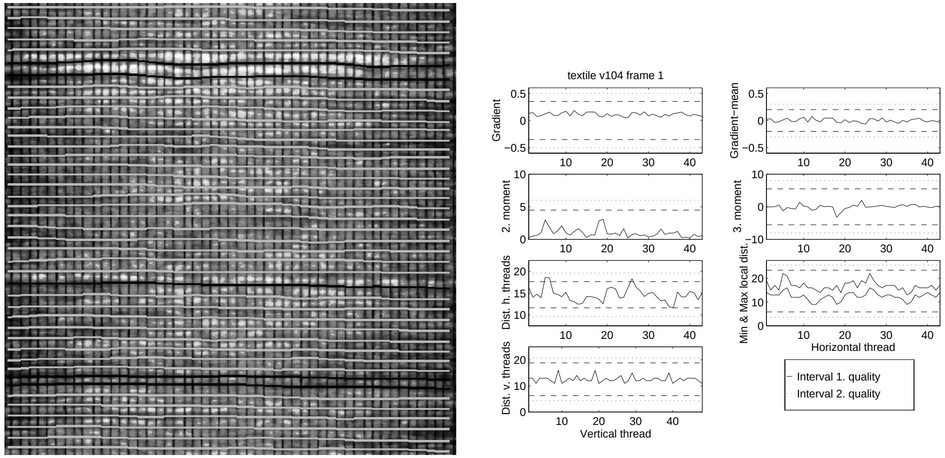


Figure B.5: Textile with large deviations in the distance between the horizontal threads (left). In the textile the located threads are drawn. Plot of the features for the defect analysis (right). Note the large peaks in the plot of the mean distance between the horizontal threads, which indicate the defects.

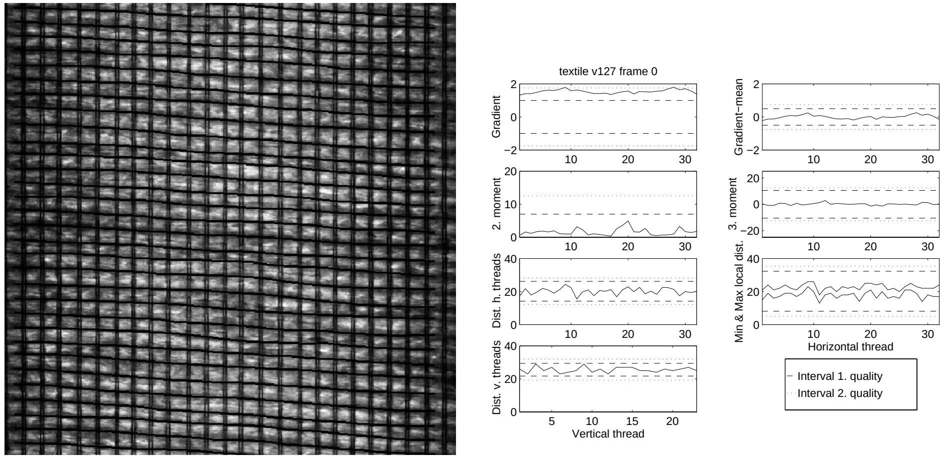


Figure B.6: Textile with oblique horizontal threads (left). In the textile the located threads are drawn. Plot of the features for the defect analysis (right). Note the high values of the gradient related to the oblique horizontal threads.

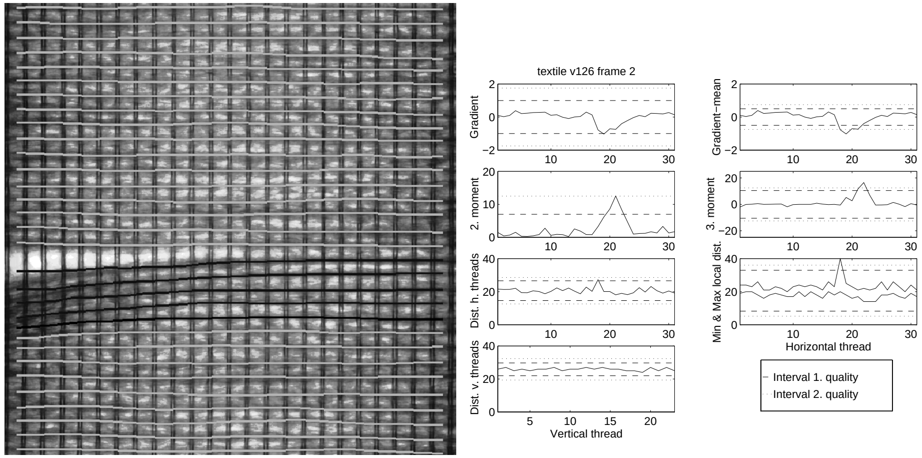


Figure B.7: Textile with oblique border of the horizontal threads (left). In the textile the located threads are drawn. Plot of the features for the analysis for defects (right). Note the peak in the 3. moment, 2. moment, and the linear variations of the gradient, which indicate the defect.

on the located horizontal threads a number of features are calculated which are used for detection of defects in the horizontal threads. All defects in the tested samples of textile have been located correctly. In relation to the realtime demands this method should be accelerated by a factor 4. This seems very encouraging while it should be possible to accelerate the method by optimizing the implementation and using dedicated hardware.

Appendix C

A General Polygon-based Deformable Model for Object Recognition

Rune Fisker and Jens Michael Carstensen
Dep. of Mathematical Modelling, Tech. University of Denmark

Abstract

We propose a general scheme for object localization and recognition based on a deformable model. The model combines shape and image properties by warping a arbitrary prototype intensity template according to the deformation in shape. The shape deformations are constrained by a probabilistic distribution, which combined with a match of the warped intensity template and the image form the final criteria used for localization and recognition of a given object. The chosen representation gives the model an ability to model an almost arbitrary object. Beside the actual model a full general scheme for applying the model is proposed. The scheme includes general methods for initialization, optimization and validation. Experimental results for real data are shown. Compared to related work the proposed model and the methods for initialization and validation contains a number of interesting features and improved abilities.

Keywords: deformable templates, active contour models, active shape models, warping.

C.1 Introduction

The group of models known as deformable models have been presented under many different names, where the best known probably are active contour models (snakes), active shape models and deformable templates. One of the characteristics of deformable models is that an object embedded in an image is represented by a vector of template parameters $\boldsymbol{v} \in \Omega$, where Ω is the parameter space. Direct or under some mapping the template parameters $\boldsymbol{v}$ define the object. To make inference about the object in an image $\boldsymbol{y}$ a probabilistic distribution $P(\boldsymbol{v}|\boldsymbol{y})$ (or an equivalent energy function) is formulated such that the configuration of the template parameters, which match the object best, have the highest probability. The probabilistic distribution can be separated in two parts: the prior $P(\boldsymbol{v})$ and the likelihood $P(\boldsymbol{y}|\boldsymbol{v})$. The prior distribution $P(\boldsymbol{v})$ represents the prior knowledge about the object and is independent of the image, and the likelihood $P(\boldsymbol{y}|\boldsymbol{v})$ (or the observation model) represents the interactions with the image (the observations). Using Bayes theorem $P(\boldsymbol{v}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v})$ the prior and likelihood is combined to the final posterior distribution used to make interference.

C.2 Deformable models

Jain et al. [102] divide deformable models into two groups: Free form and Parametric. Free form deformable templates have no explicit global structure because the prior only contains local continuity and smoothness constraints. This makes free form templates able to represent an arbitrary shape as long as the continuity and smoothness constraints are satisfied. The best known example of free form models is active contour models or snakes originally proposed by Kass et al. [108]. To compensate for the shrinking effect of closed active contour models Cohen and Cohen [33, 34] add an inflation force to the snake model. Another example is Lai and Chin [118], who proposed a more generalized formulation of active contour models.

In parametric deformable models prior knowledge of the global structure is included using a parameterized template. One of the pioneers of deformable models is Grenander, who proposes a parametric model [85] based on a 2D vector cycle representation of the given object and a stochastic distribution that governs the deformations of the initial template. This model has been further developed and described in [86, 155]. Jain et al. [102] propose a general framework for object matching based on prototype (or average) templates and with a different type of deformations and local constrains on these deformations. This model has for example been used for recognition of handwritten digits [103]. A similar model has been proposed by Garrido and Blanca [72], but they use different methods for initialization and optimization. Another general stochastic model is proposed by Staib et al. [163]. This model uses elliptic Fourier descriptors to represent the boundary of the object. Cootes et al. [41] propose an active shape model, where the object is represented by the mean shape of a training set and a linear combination of eigenvectors of the shape variation from this mean. This work has further been developed into the novel Active Appearance models [35], which also incorporates the appearance, i.e. grey level information. All the previous cited parametric models are general, because they, given a prototype template or a training set, are able to handle an almost arbitrary object. Probably the largest group of deformable models are formulated and tuned for a specific object, because it is often needed to incorporate special assumption to be able to make inference about a specific object. The properly best known model formulated for a specific problem is an eye model and a mouth model proposed by Yuille et al. [189], which is used to locate the eyes and mouth in face images. A description of more models can be found in the excellent book on Active Contours by Blake and Isard [18] and in the survey paper by McInerney and Terzopoulos [132] and by Jain et al. [101].

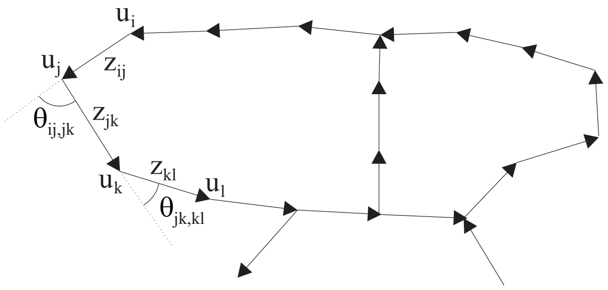


Figure C.1: The object shape represented with vertices $\boldsymbol{u_i}$ and edges $\boldsymbol{z_{ij}}$.

C.3 The polygon-based model

An object is defined by the set of template parameters $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{z})$, where $\boldsymbol{u}$ is an ordered set of vertices $\boldsymbol{u} = (\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u_{L-1}})$ and $\boldsymbol{z}$ is a set of edges $\boldsymbol{z} = (\boldsymbol{z_{ij}}, ..., \boldsymbol{z_{kl}})$, where $\boldsymbol{u_i} \in R^N$ and $\boldsymbol{z_{ij}} = \boldsymbol{u_j} - \boldsymbol{u_i}$, see figure C.1. The model can not only represent a single closed contour, but an arbitrary combination of combined or not combined polygons and lines, hence the name a polygon-based deformable model.

C.3.1 Prior model

The basic principle of prior models is to introduce prior knowledge into the model. The attraction of using prior knowledge is simply that it is so hard to make progress without it [18]. We propose a new prior model containing two parts. The first part favors configurations of the templates, where the ratio $\gamma_{ij,jk}$ between the length of edge $\boldsymbol{z}_{ij}$ and $\boldsymbol{z}_{jk}$ is close to the prior knowledge about this ratio $\gamma^{p}_{ij,jk}$. The second part penalizes the deviation between the actual angle $\theta_{ij,jk} \in [-\pi, \pi]$ and the prior angle $\theta^{p}_{ij,jk} \in [-\pi, \pi]$ between the edge $\boldsymbol{z_{ij}}$ and the edge $\boldsymbol{z_{jk}}$, see figure C.1. The prior angle $\theta^{p}_{ij,jk}$ corresponds to the prior knowledge about the value of this angle. During the formulation of the prior model focus was on incorporating model properties, which are easy to interpret, in order to make it easier to control and understand the influence of the prior. The prior model is defined as:

$$
\begin{aligned}
P(\boldsymbol{v}) &= \frac{1}{Z} exp\{-\sum_{\boldsymbol{\Psi}} \alpha^{\gamma}_{ij,jk}(\frac{|z_{ij}|}{|z_{jk}|} - \gamma^{p}_{ij,jk}))^2 \\
&\quad -\sum_{\boldsymbol{\Psi}} \alpha^{\theta}_{ij,jk}(1 - cos(\theta_{ij,jk} - \theta^{p}_{ij,jk}))\} \qquad \text{(C.1)}
\end{aligned}
$$

where $\boldsymbol{\Psi} = \{(ij, jk), ...\}$ is the set of edge pairs, where prior knowledge should be included, $|\boldsymbol{z_{jk}}|$ is the length of $\boldsymbol{z_{jk}}$, $\alpha^{\gamma}_{ij,jk} \in R_{+}$ is the prior weight parameter, which determines the relative influence of the deviation between $\frac{|z_{ij}|}{|z_{jk}|}$ and $\gamma^{p}_{ij,jk}$, $\alpha^{\theta}_{ij,jk} \in R_{+}$ is the equivalent parameter for $\theta_{ij,jk}$ and $\frac{1}{Z}$ is the normalizing constant. Note that the order of vertices indexes i,j and k in e.g. $\theta_{ij,jk}$ is arbitrary as long as one vertices always is present twice. The actual values of $\gamma^{p}_{ij,jk}$ and $\theta^{p}_{ij,jk}$ for an given object can either be determined from a training set or chosen based on empirical experience. The criterion $(1 - cos(\theta_{ij,jk} - \theta^{p}_{ij,jk}))$ is chosen to ensure the result is zero when the error is zero and to get a proper symmetric function for the radial error.

The trigonometric term in the second part of the proposed prior (C.1) can be rewritten to become a direct function of the vertices $\boldsymbol{u}$. In the case of $N = 2$, i.e. $\boldsymbol{u_i} = (x_i, y_i)$, the trigonometric term becomes:

$$
\begin{aligned}
& cos(\theta_{ij,jk} - \theta^{p}_{ij,jk}) \\
= \; & cos(\theta_{ij,jk})c^{p}_{ij,jk} + sin(\theta_{ij,jk})s^{p}_{ij,jk} \\
= \; & \frac{(x_j - x_i)(x_k - x_j) + (y_j - y_i)(y_k - y_j)}{|\boldsymbol{z_{ij}}||\boldsymbol{z_{jk}}|} c^{p}_{ij,jk} + \\
& \frac{(x_j - x_i)(y_k - y_j) - (y_j - y_i)(x_k - x_j)}{|\boldsymbol{z_{ij}}||\boldsymbol{z_{jk}}|} s^{p}_{ij,jk} \qquad \text{(C.2)}
\end{aligned}
$$

where $c^{p}_{ij,jk} = cos(\theta^{p}_{ij,jk})$, $s^{p}_{ij,jk} = sin(\theta^{p}_{ij,jk})$. Equivalent results can be derived for models of higher order ($N > 2$).

The proposed prior is able to model objects containing smooth parts and parts with corners by choosing $\theta^{p}_{ij,jk} = 0$ and $\theta^{p}_{ij,jk} = \frac{\pi}{2}$, respectively. By choosing $\theta^{p}_{ij,jk} = 0 \ \forall \ (ij, jk) \in \Psi$ the behaviour of the prior becomes very similar to the behaviour of the internal energy of a snake [108] without the shrinking effect, because the first term in (C.1) do not shrink the model and the second term in (C.1) will have similar properties as the second derivative in the snake internal energy. In general the proposed prior is able to model an arbitrary object, which can be approximated by vertices and edges. Basically most of the priors of the other general deformable models are able to model the same objects, but the proposed prior is the only model, where it is possible to control the actual value of the angle between edges and of the relative length of the edges independently. Another advantage of the proposed prior is that it is invariant to scale, rotation and translation, which is not a common feature for most of the general models. The invariance is due to the fact that both measures, i.e. $\gamma_{ij,jk}$ and $\theta_{ij,jk}$, used in the prior are relative measures, which make them invariant to scale, rotation and translation.

C.3.2 The Likelihood function

The proposed likelihood is based on the matching of an intensity template $\boldsymbol{m_v}$ to the image. The intensity template is directly connected to the template parameters $\boldsymbol{v}$, and a prototype intensity template $\boldsymbol{m_v^{pt}}$ corresponding to the parameters $\boldsymbol{v^{pt}}$ is assumed to be known. The actual configuration of the intensity template $\boldsymbol{m_v}$ corresponding to the template parameters $\boldsymbol{v}$ is determined by a warp of $\boldsymbol{m_v^{pt}}$ to $\boldsymbol{m_v}$ according to the deformation of $\boldsymbol{v^{pt}}$ to $\boldsymbol{v}$. The intensity template is only defined in a band $\boldsymbol{b_{jk}}$ around each edge $\boldsymbol{z_{jk}}$, see figure C.2. For $N = 2$ each band $\boldsymbol{b_{jk}}$ is defined by the quadrangle spanned by the four intersection points $\boldsymbol{u_{ij,jk}}^l, \boldsymbol{u_{ij,jk}}^r, \boldsymbol{u^l_{jk,kl}}$ and $\boldsymbol{u^r_{jk,kl}}$, where $\boldsymbol{u^l_{ij,jk}}$ and $\boldsymbol{u^r_{ij,jk}}$ corresponds to the intersection of the borders of band $\boldsymbol{b_{ij}}$ and $\boldsymbol{b_{jk}}$, see figure C.2. These points can be defined in two ways depending on if the width of the band is assumed to be constant or scale with the size of the template. This assumption should be based on the actual behaviour of the object, i.e. does the width of the intensity template change for different sizes of the object. The intersection points are then defined by:

$$\begin{aligned}\boldsymbol{u^l_{ij,jk}} &= \boldsymbol{u_j} + d^l_{ij,jk} tan(\frac{\theta_{ij,jk}}{2})\frac{\boldsymbol{z_{jk}}}{|\boldsymbol{z_{jk}}|} - d^l_{ij,jk}\frac{\overline{\boldsymbol{z}}_{\boldsymbol{jk}}}{|\boldsymbol{z_{jk}}|}\\ \boldsymbol{u^r_{ij,jk}} &= \boldsymbol{u_j} - d^r_{ij,jk} tan(\frac{\theta_{ij,jk}}{2})\frac{\boldsymbol{z_{jk}}}{|\boldsymbol{z_{jk}}|} + d^r_{ij,jk}\frac{\overline{\boldsymbol{z}}_{\boldsymbol{jk}}}{|\boldsymbol{z_{jk}}|}\end{aligned}$$

where $\overline{\boldsymbol{z}}_{\boldsymbol{jk}} = (-y_k + y_j, x_k - x_j)$ is the orthorgonal vector to $\boldsymbol{z_{jk}}$, $\theta_{ij,jk} = cos^{-1}(\boldsymbol{z_{ij}} \cdot \boldsymbol{z_{jk}}/(|\boldsymbol{z_{ij}}||\boldsymbol{z_{jk}}|))$, $d^l_{ij,jk} = d^l$ if the bands are assumed to have constant width or $d^l_{ij,jk} = d^l/2(|\boldsymbol{z_{ij}}|/|\boldsymbol{z^{pt}_{ij}}|+|\boldsymbol{z_{jk}}|/|\boldsymbol{z^{pt}_{jk}}|)$ if the width is assumed to vary with the size, where d^l is a given constant and $\boldsymbol{z^{pt}_{jk}}$ is the edge $\boldsymbol{z_{jk}}$ corresponding to the prototype template. The reason for choosing $d^l_{ij,jk}$ proportional to the average scaling of $\boldsymbol{z_{ij}}$ and $\boldsymbol{z_{jk}}$ for a not constant width of the band is to avoid discontinuities in the band width. The local deformation of

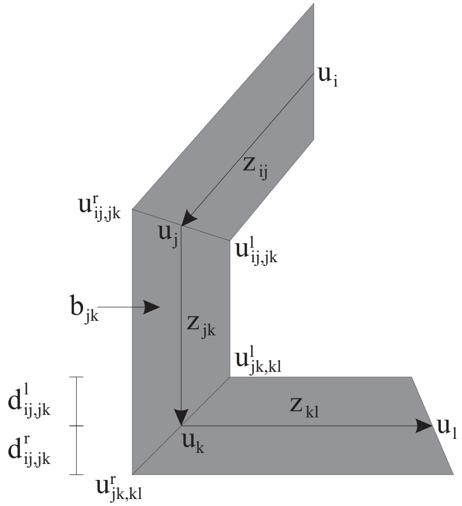


Figure C.2: The intensity template.

each band $\boldsymbol{b_{jk}}$ is approximated by a bilinear transformation:

$$\begin{aligned} x^{pt} &= r^{jk}_{10}x + r^{jk}_{01}y + r^{jk}_{11}xy + r^{jk}_{00}\\ y^{pt} &= s^{jk}_{10}x + s^{jk}_{01}y + s^{jk}_{11}xy + s^{jk}_{00} \end{aligned} \tag{C.3}$$

where (x^{pt}, y^{pt}) and (x, y) are the coordinates in the prototype template and the warped template, respectively, and $\boldsymbol{r^{jk}}$ and $\boldsymbol{s^{jk}}$ are the warping coefficients. The actual values of $\boldsymbol{r^{jk}}$ and $\boldsymbol{s^{jk}}$ are found by inserting the corresponding intersection points $\boldsymbol{u_{ij,jk}}^l, \boldsymbol{u_{ij,jk}}^r$, $\boldsymbol{u^l_{jk,kl}}, \boldsymbol{u^r_{jk,kl}}, \boldsymbol{u^{pt,l}_{ij,jk}}, \boldsymbol{u^{pt,r}_{ij,jk}}, \boldsymbol{u^{pt,l}_{jk,kl}}$ and $\boldsymbol{u^{pt,r}_{jk,kl}}$ in (C.3) and solve for $\boldsymbol{r^{jk}}$ and $\boldsymbol{s^{jk}}$. The values of (x^{pt}, y^{pt}) are real numbers, and the intensities corresponding to (x^{pt}, y^{pt}) need to be interpolated. Traditionally it is necessary to use bilinear or cubic interpolation to achieve proper results, but if the prototype template is created at least five times larger than the actual size of the template in the image, proper results can be obtained by the much faster nearest neighbour interpolation. The actual values of the intensities in the intensity template are either determined from a training set or based on empirical experience. In the special case where a point in the warped template is contained in more than one band the average, maximum or minimum value is used depending on the actual object. The equations above have been derived for the case where two edges intersects at each vertices, but the result can be generalized to any positive number of intersection in each vertices.

The criteria used for matching the image $\boldsymbol{y}$ and the warped template $\boldsymbol{m_v}$ is correlation, which leads to the following likelihood function $P(\boldsymbol{y}|\boldsymbol{v})$:

$$P(\boldsymbol{y}|\boldsymbol{v}) = \frac{1}{Z} exp\{\frac{(\sum_{r,c}(\boldsymbol{m_v}(r,c) - \mu_m)(\boldsymbol{y}(r,c) - \mu_y)}{\lambda \sigma_m \sigma_y}\} \qquad (C.4)$$

where $\sum_{r,c}$ is the sum over the N_m pixels in the template, which are not "don't cares", i.e. values outside the bands, and the corresponding image pixels, $\mu_m = \frac{1}{N_m}\sum_{r,c}\boldsymbol{m_v}(r,c)$, $\mu_y = \frac{1}{N_m}\sum_{r,c}\boldsymbol{y}(r,c)$, $\sigma_m^2 = \frac{1}{N_m}\sum_{r,c}(\boldsymbol{m_v}(r,c) - \mu_m)^2$, $\sigma_y^2 = \frac{1}{N_m}\sum_{r,c}(\boldsymbol{y}(r,c) - \mu_y)^2$ and λ is regularization constant, which gives the relative influence of the likelihood. One advantage of using correlation for matching is that it is invariant to linear transformation of the intensity. Another advantages is that the output always is between -1 and 1, which make it much easier to perform a robust thresholding. If speed is a concern and the intensity variation between objects are low computationally cheaper matching approaches like the sum of squared differences $\sum_{r,c}(\boldsymbol{m_v}(r,c) - \boldsymbol{y}(r,c))^2$ or unnormalized correlation $\sum_{r,c}\boldsymbol{m_v}(r,c)\boldsymbol{y}(r,c)$ can be used. It is approximately twice as fast to compute $\sum_{r,c}(\boldsymbol{m_v}(r,c) - \boldsymbol{y}(r,c))^2$ or $\sum_{r,c}\boldsymbol{m_v}(r,c)\boldsymbol{y}(r,c)$ as correlation.

The approach of using intensity templates in the likelihood function enable the likelihood to model an arbitrary intensity function and therefore and arbitrary object. With the exception of the work by Cootes and his colleagues [35, 41] the other general models assume the contour to be placed on a special image feature, typical the image edges. The Active Shape Models [41] and the novel Active Appearance Models [35] match a 1D gray scale profile and a eigenvector representation of a full 2D gray scale template to the image, respectively. Compared to the full intensity template the proposed band approach reduces the amount of data to warp and match. It also has the advantage of being able to focus the matching on the descriptive subparts of the object. This can be a very important feature for recognition of e.g. overlapping or transparent objects. The two other models also have the disadvantage of presuming a training set. Another difference is that for the full template approach it is not possible only to scale the intensity template a long the edges, i.e. keep the band width constant. But in general the three models each have their advantages with respect to different problems.

Finally the posterior $P(\boldsymbol{v}|\boldsymbol{y})$ is created using Bayes theorem and the prior (C.1) and the likelihood (C.4).

C.3.3 Initialization and optimization

The proposed polygon-based model is then used to make inference about an object in an image by estimating the template parameters, which corresponds to the local maximum a posteriori (MAP), defined as: $\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y})$. The MAP estimation is performed using an iterative optimization procedure, which presume an initial configuration $\boldsymbol{v^0}$ of the template parameters. This basically separates the MAP estimation in two steps: initialization and optimization. For a general review of initialization and optimization techniques used for deformable models refer to [68].

For the initialization a search strategy is used, where the general idea is to do a multiresolution search for the configurations in parameter space with the highest posterior probabilities. Traditionally the search is performed for different combinations of position, scale and rotation. The configurations with a probability above some threshold are then used as the initial configurations [102]. By a filter interpretation of the likelihood, the calculation of the likelihood can be performed for a large number of positions by a convolution of this filter and the image [68]. The result of the convolution is basically equivalent to translating the template, such that each pixel is the centre of the template, and calculating the likelihood. For a large number of models the filter interpretation of the likelihood increase the speed of the likelihood calculation significantly [68]. The filter interpretation of the likelihood for the proposed model is very easy to derived, because the likelihood filter corresponding to $\boldsymbol{v}$ is equivalent to the intensity template $\boldsymbol{m_v^{0,0}}$ with centre in $(0,0)$. Instead of a convolution of the filter and the image, the filter should be correlated with the image to calculated the likelihood. These ideas have been developed into the following algorithm for initialization:

1. Chose the image scale $\kappa \geq 1$.

2. Create the intensity templates $\boldsymbol{m_{v_o}^{0,0}}, ..., \boldsymbol{m_{v_{M-1}}^{0,0}}$ for the relevant configurations of the model $\boldsymbol{v_o}, ..., \boldsymbol{v_{M-1}}$.

3. Create the likelihood $P(\boldsymbol{y}|\boldsymbol{v_i})$ by correlating each template $\boldsymbol{m_{v_i}^{0,0}}$ with the image $\boldsymbol{y}$ and add the prior $P(\boldsymbol{v_i})$ to get the posterior $P(\boldsymbol{v_i}|\boldsymbol{y})$.

4. Extract the initial configurations, which correspond to local maxima in the posterior, and which have a value above a given threshold t_i. Add these configurations to the existing configurations.

5. Decrease the image scale $\kappa = \kappa - 1$ and optionally optimize and validate the existing configurations if $\kappa > 1$.

6. If $\kappa \geq 1$ goto 2.

The final optimization after step 6 and the optional optimization in step 5 is basically just an ordinary optimization problem, but the optimization is not trivial, due the dimensionality and multimodality. A number of general gradient and non-gradient based optimization

algorithms have been applied and the most successful, in the sense of achieving the best average probabilities, seems to be the non-gradient based optimization method Pattern Search [94].

C.3.4 Validation

For non-trivial object recognition tasks, e.g. object with high variability in shape and intensity, and for a coarse sampling of the relevant configuration of the initial template (in step 2) a number of false initial candidates might be accepted. Even after the optimization a number of these false candidates might be caught in local maxima, which do not correspond to a true object. To remove the false objects a validation might need to be performed. A simple validation is performed by removing all optimized templates, which have a probability below a given threshold $t_v > t_i$. This validation can also be used to determine whether one or more objects of a given type is present in the image.

For a non-trivial task the simple validation might not be good enough, because subparts of objects gives a high response or two small objects look like one big etc. This requires an additional validation of overlapping templates to remove false candidates. In practice the validation is performed by examining all templates, which have their centre inside another template. If the probability of the template, which is inside, is lower than the other template, the first template is removed. This method does not guarantee that true objects are removed, but in general it tend to remove more false objects than true object, because of the comparison of energy. One obvious limit of this validation is that if two true objects overlap more than approximately 50 percent, one true object will be removed.

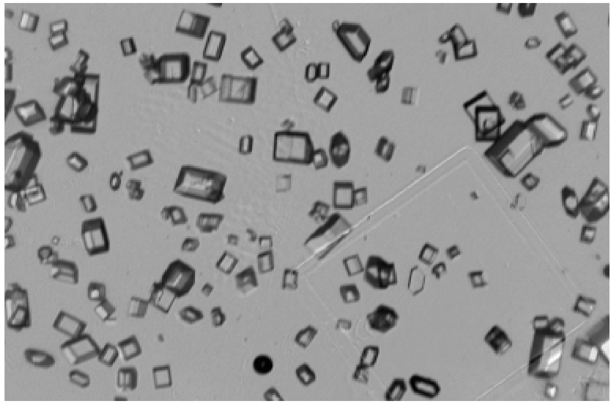


Figure C.3: Example of pre-processed image of sugar crystals.

C.4 Experimental Results

The polygon-based deformable model have been applied to a real problem, where sugar crystals should be located and their size measured in images, which are acquired during the crystallisation process of sugar, see figure C.3. The size distribution of the sugar crystals is an important feature for the final quality of sugar, and the producers of sugar are very interested in measuring this property during the crystallisation to control the distribution. At first glance the task seems quite trivial, but due to the high variability in shape and local intensity distribution, missing parts and the tendency to touch and overlap, simpler approaches based on grey scale morphology, thresholding and blob analysis have failed.

Based on empirical experience the individual crystals are assumed to be modelled by a simple polygon model with four vertices $\boldsymbol{u_0}, \boldsymbol{u_1}, \boldsymbol{u_2}$ and $\boldsymbol{u_3}$ and four edges $\boldsymbol{z_{0\,1}}, \boldsymbol{z_{1\,2}}, \boldsymbol{z_{2\,3}}$ and $\boldsymbol{z_{3\,0}}$. The ideal crystal is assumed to be an rectangle, which leads to the prior assumption $\theta^p_{ij,jk} = \pi/2 \;\forall\; (ij,jk) \in \Psi$, where $\Psi = \{(0\,1, 1\,2), (1\,2, 2\,3), (2\,3, 3\,0), (3\,0, 0\,1)\}$. The values for $\gamma^p_{ij,jk}$ are chosen individually such that the

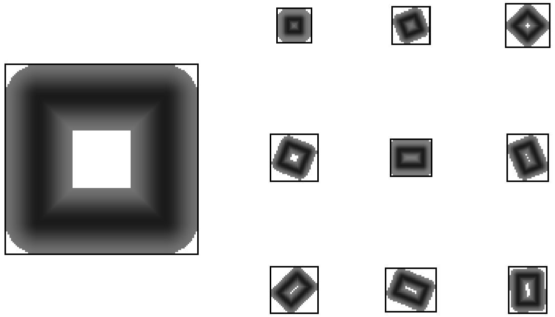


Figure C.4: Prototype intensity template (left) and filter bank used for initialization at full image size, i.e. $\kappa = 1$ (right). White values inside the filter indicates "don't cares", i.e. values outside the bands.

prior probability of the initial configuration always is zero. Based on empirical observations the intensity orthogonal to the edges is approximated by a Gaussian distribution with the mean at the edges and the variance σ_I^2. This assumption is used for the creation of the prototype intensity template, see figure C.4. The width of the crystal borders seems to be relative independent of the crystal size, so constant values are chosen for $\sigma_I^2 = 2$ and for the width $d^l_{jk} = 2{*}\sigma_I$ of the intensity band $\boldsymbol{b_{jk}}$. The model weight parameters are chosen to $\alpha^\gamma_{ij,jk} = 0.05$, $\alpha^\theta_{ij,jk} = 1 \forall\ (ij,jk) \in \Psi$ and $\lambda = 1$.

The initailization is performed according to the proposed algorithm starting at $\kappa = 2$, i.e. the original image is reduced to half the width and height. A bank of intensity template filters $\boldsymbol{m^{0,0}_{v_o}}, ..., \boldsymbol{m^{0,0}_{v_{M-1}}}$ are created for each image scale. In practice the different templates $\boldsymbol{v_o}, ..., \boldsymbol{v_{M-1}}$ are created from a default rectangle and a default square, which are scaled on a given interval and rotate $5\pi/8$ radians for each configuration, see figure C.4. Note, that the filters show the warp of the prototype intensity template according to dif-

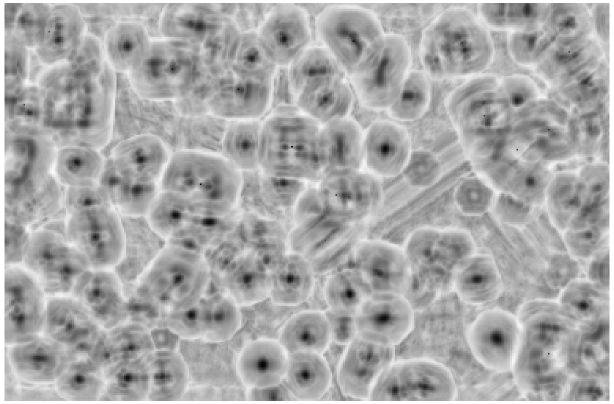


Figure C.5: The maximum posterior image $I_{max(P(\boldsymbol{v_i}|(\boldsymbol{y}))}$ at full image size. Dark correspond to low energy, i.e. high probability .

ferent configurations of the template parameters, and that crystals which are smaller that the smallest filter template, in general not are of interest. The filters are then correlated with the image and the prior energies are added. The result is M posterior energy images $I_{P(\boldsymbol{v_0}|\boldsymbol{y})}, ..., I_{P(\boldsymbol{v_{M-1}}|\boldsymbol{y})}$, where the pixel $I_{P(\boldsymbol{v_i}|\boldsymbol{y})}(r, c)$ is identical to the posterior $P(\boldsymbol{v_i}|\boldsymbol{y})$ for the template $\boldsymbol{v_i}$ with centre in (r, c). Note, that in practise the energies equivalent to $-log$ of the probability is always used and not the actual probabilities. To extract the local energy minima (equivalent to the probabilistic maxima) the maximum posterior image $I_{max(P(\boldsymbol{v_i}|(\boldsymbol{y}))} = max(I_{P(\boldsymbol{v_0}|\boldsymbol{y})}, ..., I_{P(\boldsymbol{v_{M-1}}|\boldsymbol{y})})$ is created, see figure C.5. The dark spots in the figure indicate positions, which correspond to a crystal in the image. Note, that only the smaller crystals show large response, because only the small initial templates are used at this resolution. The large templates are initialized at a lower resolution.

The configurations corresponding to local minima below the threshold $t_i = -0.4$ are extracted as initial configurations. If other templates already exist from an initialization at a higher scale, the posterior energy of these templates are inserted in the maximum posterior image at the pixel, which correspond to the centre of each template. These energies appear as black dots in the maximum posterior image, see figure C.5. The energies are inserted before the extraction of new initial configurations. During the multiresolution initialization the optional optimization and validation is performed to remove false candidates at an early point and to increase reliability of the energies of the existing templates. The result of the initialization can be seen in figure C.6. Note that templates on dark objects, which touch the boundary of the image, are removed to avoid dealing with boundary conditions. The result of the final initialization contains a number of false candidates, but in general it is better to accept a number of false candidates than to reject a number of true objects, because false candidates usually will be removed during the validation. In general the exact choice of the threshold t_i and of the number M of configuration used for initialization is a trade-off between speed and accuracy, because a lower threshold t_i or an increased number of configuration M will improve the accuracy of the final result, but also increase the computational cost.

The initial configurations are fed to Pattern Search, which optimize the individual templates, see figure C.7. The optimized result is then validated with respect to energy and overlap to obtain the final result, see figure C.8. The validation removes a lot of false candidates, such that the final result with few exceptions contains true objects. Unfortunately the overlap validation also removes a few overlapping true objects. From a general point of view it is not satisfying that objects are missed, but for the actual application it is important that the accepted objects corresponds to true objects, and it does not matter that a few crystal are missed as long as it is not systematic (what is not the experience). In general the method recognise a large percentage of the crystals, including crystals where pieces of the border are missing and crystal which are touching and partly overlapping.

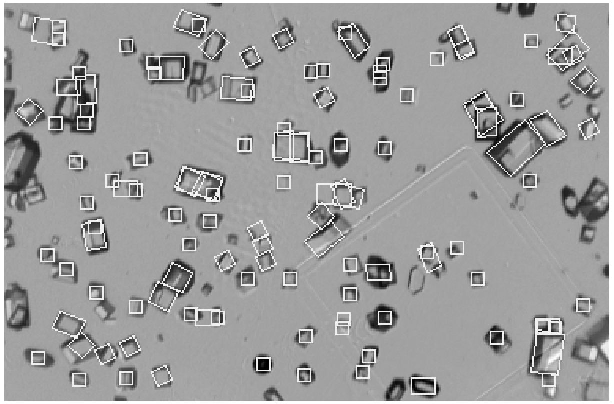


Figure C.6: The result of the initialization.

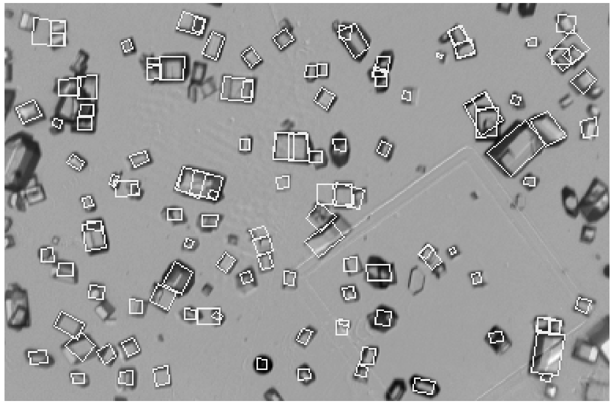


Figure C.7: The result after the optimization.

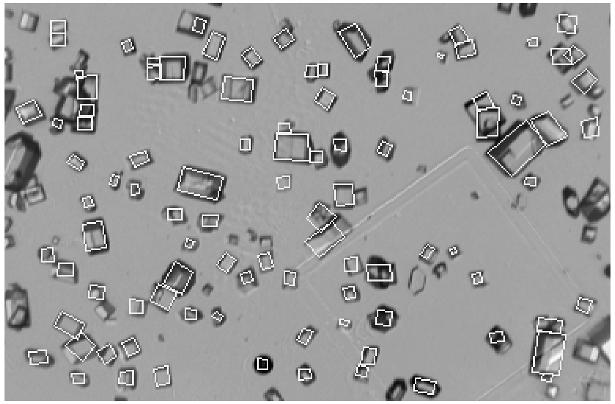


Figure C.8: The final result after validation.

C.5 Discussions and future work

This work presents a systematic method for object localization and recognition. The framework is general in the sense that it can be applied to almost any object. Promising experimental results have been presented on an object recognition task, where the objects have a high variability in appearance, have missing parts and a tendency to touch and overlap. The model and the scheme have been applied to 2D greyscale images, but can easily be generalized to higher dimensions and multiband images.

We are in the process of improving the validation of overlapping objects and incorporate general local measures to accelerate the initialization, which hopefully will lead to an further improved scheme. Regarding the actual sugar crystal data, improvement should be obtained by creating a more advanced intensity template rather than improving the model or the scheme. At the moment the model is also considered used for a number of other problems.

Appendix D

Extending and applying Active Appearance Models for automated, high precision segmentation in different image modalities

Mikkel B. Stegmann, Rune Fisker, Bjarne K. Ersbll
Dep. of Mathematical Modelling, Tech. University of Denmark

Abstract

In this paper, we present a set of extensions to the deformable template model: Active Appearance Model (AAM) proposed by Cootes et al. AAMs distinguish themselves by learning a priori knowledge through observation of shape and texture variation in a training set. This is used to obtain a compact object class description, which can be employed to rapidly search images for new object instances. The proposed extensions concern enhanced shape representation, handling of homogeneous and heterogeneous textures, refinement optimization using Simulated Annealing and robust statistics. Finally, an initialization scheme is designed thus making the usage of AAMs fully automated. Using these extensions it is demonstrated that AAMs can segment bone structures in radiographs, pork chops in perspective images and the left ventricle in cardiovascular magnetic resonance images in a robust, fast and accurate manner. Subpixel landmark accuracy was obtained in two of the three cases.

Keywords: Deformable Template Models, Snakes, Robust Statistics, Initialization, Metacarpal Radiographs, Cardiovascular Magnetic Resonance Imaging, Segmentation.

D.1 Introduction

In recent years, the model-based approach towards image interpretation named deformable template models has proven very successful. This is especially true in the case of images containing known objects with large variability.

Among the earliest and most well known deformable template models is the Active Contour Model – known as Snakes proposed by Kass et al. [108]. Snakes represent objects as a set of outline landmarks upon which a correlation structure is forced to constrain local shape changes. In order to improve specificity, many attempts at hand crafting a priori knowledge into a deformable template model have been carried out. These include parameterization of a human eye using ellipsis and arcs by Yuille et al. [189].

In a more general approach, while preserving specificity Cootes et al. [41] proposed the Active Shape Models (ASM) where shape variability is learned through experimental observation. In practice, this is accomplished by a training set of annotated examples followed by a Procrustes analysis combined with a principal component analysis.

A direct extension of the ASM approach has lead to the Active Appearance Models (AAMs) [35]. Besides shape information, the textural information, i.e. the pixel intensities across the object, is included into the model. AAMs have been further developed in [39, 40, 61, 136]. Quite similar to AAMs and developed in parallel herewith, Sclaroff & Isidoro suggested the Active Blob approach [156]. Active Blobs is a real-time tracking technique, which captures shape and textual information from a prototype image using a finite element model, to model shape variation. Compared to AAMs, Active Blobs deform a static texture, whereas AAMs deforms both texture and shape during the optimization. Early modeling of texture includes the eigenfaces by Turk & Pentland [174], where face recognition was accomplished using a PCA-based texture model similar to the one integrated into AAMs.

Other general deformable template models include the ones proposed by Grenander [85] and Jain [102]. For further information on deformable template models, refer to the surveys given in [63, 101, 132].

D.2 Active Appearance Models

Below is presented the outline of the Active Appearance Model approach. AAMs distinguish themselves from many other segmentation methods in the sense that segmentation can be carried out using the approach as a black box. The user only needs to provide a training set of annotated shapes. For further details refer to [35, 40, 164].

D.2.1 Shape & Landmarks

AAMs handle shapes as a finite set of landmarks. Here the term shape is defined as "All the geometrical information that remains when location, scale and rotational effects are filtered out from an object." [56] and the concept of a landmark as "A point of correspondence on each object that matches between and within populations." [56].

A mathematical representation of a shape with n-points in k dimensions could be a concatenation of each dimension in a kn-vector. The vector representation used for planar shapes is then:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^{\mathrm{T}} \tag{D.1}$$

Notice that the above representation does not contain any explicit information about point connectivity. In the presented framework, point connectivity is added as auxiliary data.

D.2.2 Shape Formulation

When dealing with redundancy in multivariate data – such as shapes – AAMs utilize the linear orthogonal transformation; principal component analysis (PCA). In our application for describing shape variation by PCA – a shape of n points is considered one observation, $\boldsymbol{x}_i$, in a $2n$ dimensional space.

In practice the PCA is performed as an eigenanalysis of the covariance matrix of the shapes aligned w.r.t. position, scale and rotation, i.e. after a Procrustes analysis. As shape metric in the alignment procedure the Procrustes distance [83] is used. Other shape metrics such as the Hausdorff distance [97] could also be considered.

Consequently it is assumed that the set of N shapes constitutes some ellipsoid structure of which the centroid – the mean shape – can be estimated as: $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The ML estimate of the covariance matrix can thus be given as, $\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}}$ The principal axes of the $2n^{th}$ dimensional shape ellipsoid are then given as the eigenvectors, $\mathbf{\Phi}_s$, of the covariance matrix, $\mathbf{\Sigma}$ (where $\mathbf{\Lambda}_s$ is a diagonal matrix of eigenvalues):

$$\mathbf{\Sigma}\mathbf{\Phi}_s = \mathbf{\Phi}_s\mathbf{\Lambda}_s \tag{D.2}$$

A new shape instance can then be generated by deforming the mean shape by a linear combination of eigenvectors, weighted by $\boldsymbol{b}_s$:

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}_s\boldsymbol{b}_s \tag{D.3}$$

Essentially, the point or nodal representation of shape has now been transformed into a modal representation where modes are ordered according to the percentage of variation that they explain. To regularize and improve performance modes are included until the cumulated variation is above a certain threshold (e.g. 95%).

D.2.3 Texture Formulation

Contrary to the prevalent understanding of the term texture in the computer vision community, this concept will be used somewhat differently below. Here we define texture as ”The pixel intensities across the object in question (if necessary after a suitable normalization).” A vector is chosen, as the mathematical representation of texture, where m denotes the number of pixel samples over the object surface:

$$\boldsymbol{g} = (g_1, g_2, \ldots, g_m)^{\mathrm{T}} \tag{D.4}$$

In the shape case, the data acquisition is straightforward because the landmarks in the shape vector constitute the data itself. In the texture-case one needs a consistent method for collecting the texture information between the landmarks, i.e. an image sampling

function needs to be established. This can be done in several ways. Here, a piece-wise affine warp based on the Delaunay triangulation of the mean shape is used. Alternatively thin-plate splines [19] could substitute the piece-wise affine warp to obtain a smooth warp. For details on the Delaunay triangulation and image warping refer to [80, 161].

Following the warp from an actual shape to the mean shape, a normalization of the $\boldsymbol{g}$-vector set is performed to avoid the influence from global linear changes in pixel intensities. Hereafter, the analysis is identical to that of the shapes. Hence, a compact PCA representation is derived to deform the texture in a manner similar to what is observed in the training set:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{\Phi}_g\boldsymbol{b}_g \tag{D.5}$$

Where $\overline{\mathbf{g}}$ is the mean texture; $\mathbf{\Phi}_g$ represents the eigenvectors of the covariance matrix and finally $\boldsymbol{b}_g$ are the modal texture deformation parameters.

For all practical purposes there will always be far more dimensions in the texture vectors than observations (annotated examples) thus leading to rank deficiency in the covariance matrix. Hence, to efficiently compute the eigenvectors of the covariance matrix one must reduce the problem through use of the Eckart-Young theorem.

D.2.4 Combined Model Formulation

To remove correlation between shape and texture model parameters – and to make the model representation even more compact – a 3rd PCA is performed on the shape and texture PCA scores of the training set, $\boldsymbol{b}$ to obtain the combined model parameters, $\boldsymbol{c}$:

$$\boldsymbol{b} = \boldsymbol{Q}\boldsymbol{c} \tag{D.6}$$

The PCA scores are easily obtained due to the linear nature of the model:

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{W}_s \boldsymbol{b}_s \\ \boldsymbol{b}_g \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}_s \boldsymbol{\Phi}_s^{\mathrm{T}} (\boldsymbol{x} - \overline{\boldsymbol{x}}) \\ \boldsymbol{\Phi}_g^{\mathrm{T}} (\boldsymbol{g} - \overline{\boldsymbol{g}}) \end{pmatrix} \tag{D.7}$$

A suitable weighting between pixel distances and pixel intensities is obtained through the diagonal matrix $\boldsymbol{W}_s$ [40]. An alternative approach is to perform the two initial PCAs based on the correlation matrix as opposed to the covariance matrix.

Now, a complete model instance including shape, $\boldsymbol{x}$ and texture, $\boldsymbol{g}$, is generated using the $\boldsymbol{c}$-model parameters.

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{\Phi}_s \boldsymbol{W}_s^{-1} \boldsymbol{Q}_s \boldsymbol{c} \tag{D.8}$$

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{\Phi}_g \boldsymbol{Q}_g \boldsymbol{c} \tag{D.9}$$

Regarding the compression of the model parameters, one should notice that the rank of $\boldsymbol{Q}$ will never exceed the number of examples in the training set.

Another feasible method to obtain the combined model is to concatenate both shape points and texture information into one observation vector from the start and then perform PCA on the correlation matrix of these observations.

D.2.5 Optimization

In AAMs the search is treated as an optimization problem in which the difference between the synthesized object delivered by the AAM and an actual image is to be minimized. By adjusting the AAM-parameters ($\boldsymbol{c}$ and pose) the model texture, g_{model}, can be deformed

to fit the image, g_{image}, in the best possible way. The quadratic error norm is applied as optimization criterion [40]:

$$E = \sum_{i=1}^{m} (g_{model} - g_{image})^2 = \sum_{i=1}^{m} (\delta g_i)^2 = |\delta \boldsymbol{g}|^2 \tag{D.10}$$

Though the parameterization of the object class in question can be compressed markedly by the principal component analysis – by leaving out the principal axes that explain little variation – it is far from an easy task to optimize the system. This is not only computationally cumbersome but also theoretically challenging since it is not guaranteed that the search-hyperspace is convex. However, AAMs circumvent these potential problems in a rather untraditional fashion, assuming a linear relationship between parameter changes, $\delta \boldsymbol{c}$, and pixel differences, $\delta \boldsymbol{g}$.

$$\delta \boldsymbol{c} = \boldsymbol{R} \delta \boldsymbol{g} \tag{D.11}$$

Since the matrix $\boldsymbol{R}$ is estimated once at model building time, this is very run-time efficient by avoiding any computationally expensive and potentially unstable high-dimensional optimization. In practice $\boldsymbol{R}$ is estimated by a set of experiments using the training set, which are fed into a multivariate principal regression framework. In the AAM optimization, this prediction scheme is applied iteratively. Fig. D.1 shows a prediction plot for one pose parameter. For details refer to [40, 164]. It should be noticed that the Active Blobs approach [156] is optimized using a method quite similar to that of AAMs named difference decomposition [81].

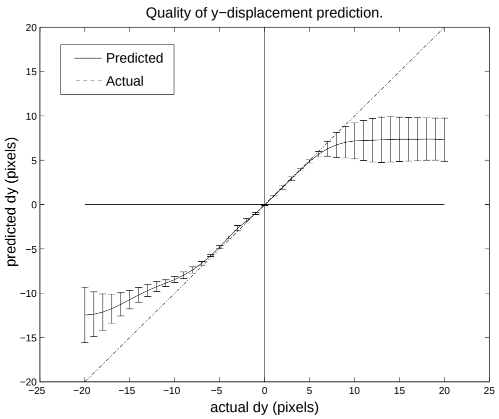


Figure D.1: Displacement plot for a series of model predications versus the actual displacement. Error bars are equal to one std.dev.

D.3 Extensions

D.3.1 Enhanced Shape Representation

Basic AAMs using the piece-wise affine warping, rely on the Delaunay triangulation of the shape points. This results in a triangular mesh covering the convex hull of the point set. For concave shapes this might not be the optimal solution. For example there could be substantial texture noise in the area outside the actual shape – but still inside the convex hull – thus leading to a less compact model.

To avoid this we suggest removing the triangles outside the shape. This is trivially accomplished by traversing the triangles; testing if the triangle centroid should be outside the shape polygon. If so, remove the triangle. To test if a point is inside the shape polygon we utilize the simple geometrical fact that, if a line from the point, $\boldsymbol{p}$, to infinity crosses the polygon an odd number of times, then the point $\boldsymbol{p}$ is inside the polygon.

Despite the above, problems remain where greater flexibility is required. Objects can contain areas where the texture variation might be considered noise. One thus wants to exclude such areas due to arguments similar to the above given. Another situation is that of having several structured objects, but in between those, the texture is highly unstructured. Features to accommodate such situations are implemented in the current AAM framework. Shapes are defined in terms of paths, which is a subset of the shape points with a given point connectivity. Each path can have a combination of the following properties:

- **Open/closed path** – Open path: a path where all points are connected to two points each.
- **Inner/outer path** – Outer path: a path where only the inside is included into the shape surface.
- **Original/artificial path** – Artificial path: a path added after the original annotation.
- **Hole/non-hole** – Hole: a path where the inside is excluded from the shape surface.

This provides a high degree of freedom, resulting in a more optimal AAM representation of the given problem. For further details, refer to [164].

D.3.2 Neighborhood AAMs

While the removal of convex triangles gave greater shape control it also increases the risk of what we coin the shrinking problem. During matching of objects with a relatively homogeneous texture, matches sometimes tend to lie inside the real object. This is due to the fact that the AAMs evaluate the fit on the object texture only.

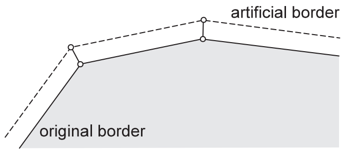


Figure D.2: Shape neighborhood added using an artificial border placed along the normal vectors of the original model points.

To avoid this we suggest including some neighboring region of the object. This will usually lead to more texture contrast in the model, since objects (often) are identified as something that stands out from its surroundings.

Neighborhood adding must be done carefully to avoid introducing new shape information. More precisely the shape PCA must retain its eigenvalue distribution. To accomplish this, we generate shape points fully correlated with the original points. The curvature is estimated at each original point. Then each new point is placed on the normal vector in a distance proportional to the relative size of the shape. See fig. D.2. However, the texture PCA will suffer since the goal is to add new information, namely background pixels, of which one could expect a substantially higher degree of variation than across the object.

The metacarpal bones in fig. D.3 serve as a good example of a shape with a relative homogeneous surface. By adding neighborhood the texture in fig. D.3 (b) is substantially more specific than the shape without, fig. D.3 (a).

D.3.3 Border AAMs

While the previous section provided a method for handling homogeneous objects, this section concerns the counter-example; heterogeneous objects.

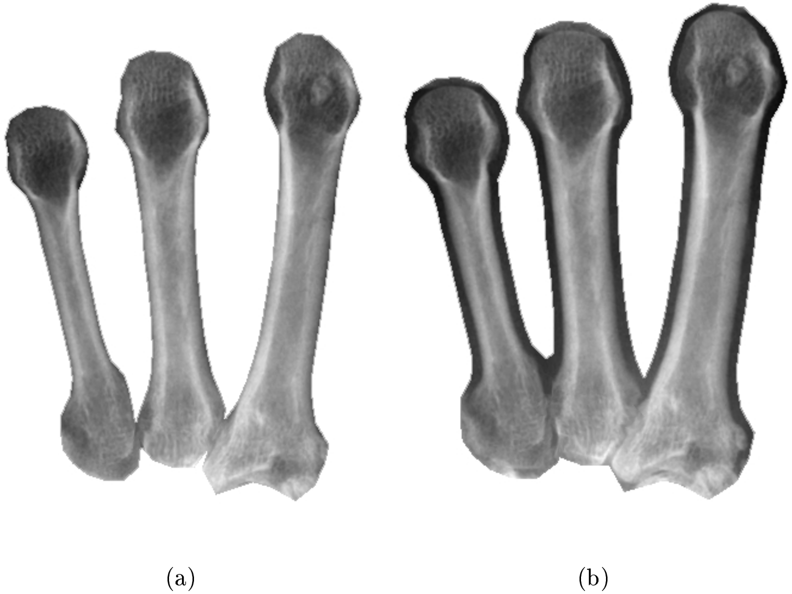


Figure D.3: (a) Shape annotated using 150 landmarks. (b) Shape with a neighborhood region added resulting in $2 \times 150 = 300$ landmarks.

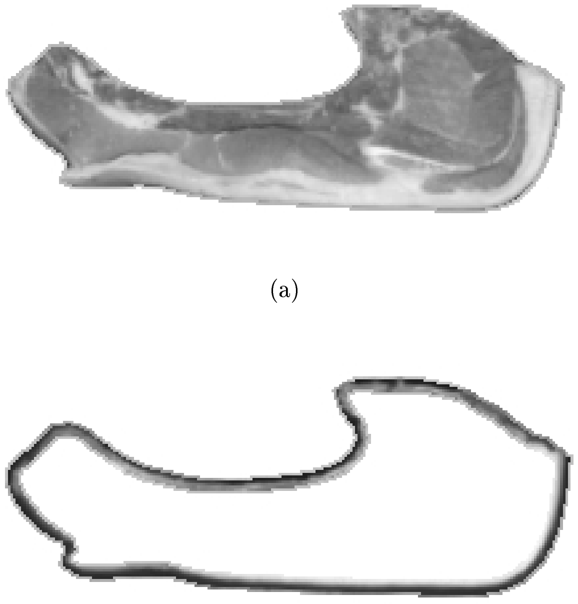
For the described texture model, it is not possible to capture objects with large heterogeneity i.e. high texture variation. Think of this as a signal with a lack of structure. In such cases, we suggest to capture only an area around the outer rim, thereby excluding the "noisy" part of the object. This approach should be feasible since we (often) perceptually identify the outer rim due to it is structured behavior (often an abrupt change in intensity). We call this a border AAM. Using the enhanced shape representation, a border AAM is simply achieved by adding an interior path, which defines a hole and by adding an outer path as described in the previous section.

By using this rationale AAMs can be made insensitive to large-scale texture noise inside the shapes, which otherwise would lead to a poor texture fit and a low landmark accuracy. The pork chops of fig. D.4 constitute a good example of this situation, due to the heterogeneity of the complex structure of fat and meat from one training example to another.

To conclude this section, we stress that border AAMs also should be substantially faster than basic AAMs, since only a fraction of the original pixels is considered.

D.3.4 Fine-tuning the model fit

The AAM search provides a fast way of optimizing the AAM using prior knowledge. However, this might not lead to the optimal solution, primarily due to weakness in the assumption that the optimization problems in an AAM search are strictly similar to those observed in a training set. Thus, we suggest fine-tuning the model fit by using a general-purpose optimization method. This approach is feasible since it is reasonable to assume that we are close to the optimum after the traditional AAM search. Hence, the optimization fine-tuning should be possible in a reasonable amount of time. Though one is not guaranteed that the hyperspace is smooth at the position where the AAM search has converged, it is still more proba-



(a)

(b)

Figure D.4: (a) Shape annotated using 83 landmarks. (b) Border shape with $3 \times 83 = 249$ landmarks.

ble that we are near a well-behaved manifold in the hyperspace. The considered optimization methods are:

- **Gradient based methods:** Steepest descent, Conjugate gradient, Quasi-Newton (BFGS) [71]
- **Non-gradient based methods:** Pattern search [94]
- **Random-sampling based methods:** Simulated annealing [27, 112]

Preliminary experiments have shown that the random-sampling based method simulated annealing has best performance. Hence, this is the only optimization method considered in the experimental section. However, investigations are not conclusive since the performance of these methods is quite sensitive to configuration parameters – i.e. step sizes, standard deviations, stop criterions etc. Nevertheless, the decision is motivated by the observation that our objective function, $|\delta \boldsymbol{g}|^2$, is most likely non-convex. Thus, deterministic optimization techniques have a high risk of being caught in spurious minima's, whereas random-sampling techniques are more likely to escape.

D.3.5 Applying Robust Statistics

As seen earlier, AAM optimization is driven by texture differences, i.e. $|\delta \boldsymbol{g}|^2$. The measure, with which the optimization evaluates itself, is here forth named the similarity measure. However, the term similar is inherently vague in a mathematical sense. This section, will dwell on interpretations of the term similar that mimics the human ability of compensating for small numbers of gross errors, thus achieving robustness in recognition. These are called robust similarity measures where the term robust refers to the insensitivity to outliers. Cootes et al. [61] previously extended the basic AAM with learning-based matching to obtain robustness. This is achieved

using a threshold for each element in $\delta \boldsymbol{g}$ estimated from the training set.

We suggest using robust similarity measures. To formalize the model fitting problem, a set of parameters, $\boldsymbol{c} = [c_1, \ldots, c_p]^{\mathrm{T}}$, are adjusted to fit a set of measurements (e.g. an image), $\boldsymbol{g} = [g_1, \ldots, g_m]^{\mathrm{T}}$. This is done by a minimization of the residuals:

$$E = \sum_{i=1}^{m} \rho(g_i - u(i, \boldsymbol{c}), \sigma_s) = \sum_{i=1}^{m} \rho(e_i, \sigma_s) \tag{D.12}$$

where u is a function that returns the model reconstruction of the i^{th} measurement and σ_s is the scale parameter that determines what should be deemed outliers. The ρ-function determines the weighting of the residuals, and is also called the error norm where the most common error norm is the quadratic norm: $\rho(e_i) = e_i^2$. This is often referred to as the L_2 norm, which is the one used by basic AAMs, see (D.10).

It is easily seen, that the quadratic norm is notoriously sensitive to outliers, since these will contribute highly to the overall solution due the rapid growth of the x^2 function. It is therefore preferable to use a norm that falls off at large residuals. As an example of a smooth norm that falls off quickly is the Lorentzian estimator [17]:

$$\rho(e_i, \sigma_s) = \log(1 + \frac{e_i^2}{2\sigma_s^2}) \tag{D.13}$$

The Lorentzian norm has been integrated into the basic AAM to supplement the quadratic norm of Cootes et al. However, one should notice that even though the AAM-search evaluate its predictions using a robust measure, the predictions themselves are done using the pixel differences directly. To address this problem Cootes et al. [61] perform a thresholding of the texture vector before the prediction. This could be viewed upon as a robust preprocessing step. The threshold limit is estimated from the training set.

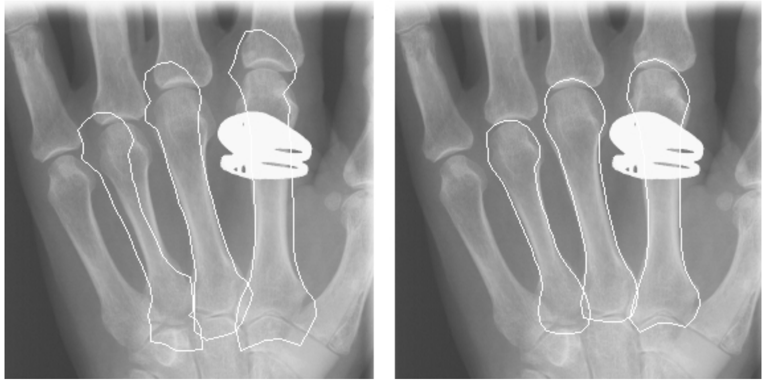


Figure D.5: Example of AAM search and Simulated Annealing fine-tuning, without (left) and with (right) the use of a robust similarity measure (Lorentzian error norm). Landmark error decreased from 7.0 to 2.4 pixels (pt.crv. error).

To demonstrate the effect of a robust error norm, an AAM search with fine-tuning using Simulated Annealing has been done with and without the Lorentzian estimator. Since radiographs are 2D projections of density, people wearing finger rings will have high-white outliers on one or more phalanges.[1] In the case given in fig. D.5 the Lorentzian error norm was used. To simulate outliers the radiograph has been manipulated so that it appears as if the metacarpal is wearing a finger ring.[2] While not perfect, the robust AAM provides a substantially better fit compared to that of the basic AAM (both using simulated annealing).

For a further treatment of robust error norms and line processes in vision refer to [17].

[1] Other outlier examples include highlights in perspective images and absence of interior parts, occlusion etc.

[2] Though this is highly unlikely since the metacarpals are situated in the middle of the hand.

D.3.6 Initialization

The basic AAM optimization scheme is inherently dependent on good initialization. To accommodate this, we devise the following search-based scheme thus making the use of AAMs fully automated. The technique is inspired by the work of Cootes et al. [61] who use a pixel difference evaluation criteria and a threshold estimation for detecting multiple object instances.

The fact that the AAMs are self-contained is exploited in the initialization – i.e. they can fully synthesize (near) photo-realistic objects of the class that they represent concerning shape and textural appearance. Hence, we use the model without any additional data to perform the initialization.

The idea is to exploit an inherent property of the AAM-optimization – i.e. convergence within some range from the optimum. See e.g. fig. D.1. This is utilized to narrow down an exhaustive search from a dense to a sparse population of the hyperspace spanned by pose- and $\boldsymbol{c}$-parameters. In other words, normal AAM-optimizations are performed sparsely over the image using perturbations of the pose and model parameters.

This has proven to be both feasible, fast and robust. A set of relevant search configuration ranges is established and the sampling within this set is done as sparsely as possible. Any available prior knowledge about pose is utilized when determining search ranges.

The crucial part of this algorithm is somewhat inspired from the class of Genetic Algorithms.[3] The total set of search configurations constitutes the initial population of candidates. From this we let the n fittest survive. These are then reproduced into more evolved guesses. From these the best is drawn and deemed the initial configuration. In pseudo-code, the initialization scheme for detecting one object per image is:

[3] Notice however, while GAs are probabilistic, our technique is deterministic. Further, are the aspects of mutation and crossover in GAs not utilized here.

```
1. Set $m$ to a suitable low number (we use $m = 3$)
2. Establish a candidate set, $\{\boldsymbol{K}\}$, containing $n$ result entries
3. Obtain application specific search ranges within each parameter (e.g. $-\sigma_1 \leq c_1 \leq \sigma_1,\ x_{min} \leq x \leq x_{max}$, etc.)
4. Populate the space spanned by the ranges – as sparsely as the linear regression allows – by a set of search configurations $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$.
5. For each vector in $\boldsymbol{V}$
6.     Do AAM optimization (max $m$ iterations)
7.     Calculate the fit, $E = |\delta \boldsymbol{g}|^2$
8.     If $E < \max_E \{\boldsymbol{K}\}$ add $(\boldsymbol{v}_i, E)$. If the number of elements in $\{\boldsymbol{K}\}$ exceeds $n$, then remove $\max_E \{\boldsymbol{K}\}$
9. End
10. For each element in $\{\boldsymbol{K}\}$
11.     Do AAM optimization (max $k$ iterations, $k > m$)
12.     Calculate and update the fit, $E = |\delta \boldsymbol{g}|^2$
13. End
```

The element in $\{\boldsymbol{K}\}$ with the minimum E will now hold the initial configuration.

We stress that the application specific search ranges in step 3 are merely a help to increase initialization speed and robustness rather than a requirement. If no prior is known, step 3 is eliminated and an exhaustive search is performed.

This scheme is readily extended into more than one object per image by a clustering of the candidate set using overlap tests. The approach

in general can be accelerated substantially by searching in a multi-resolution (pyramidal) representation of the image. For a detailed treatment of initialization of deformable template models refer to [63].

D.4 Experimental Results

Segmentation in medical images has always posed a difficult problem due to the special image modalities (CT, MRI, PET etc.) and the large biological variability. Thus, AAMs and the proposed extensions have been assessed on the three different modalities; see fig. D.6 and below:

- **A – Radiographs of Metacarpals**
  Training set: 23 images/annotations
  Image size: 240×275 pixels
  Shape model: 150 landmarks
  Texture model: $\sim$ 13.000 pixels
  95% variation explained using: 18 parameters

- **B – Cardiovascular MRIs**
  Training set: 13 images/annotations
  Image size: 256×191 pixels
  Shape model: 83 landmarks
  Texture model: $\sim$ 15.000 pixels
  95% variation explained using: 10 parameters

- **C – Perspective images of Pork Chops**
  Training set: 13 images/annotations
  Image size: 256×256 pixels
  Shape model: 66 landmarks
  Texture model: $\sim$ 2.200 pixels
  95% variation explained using: 11 parameters

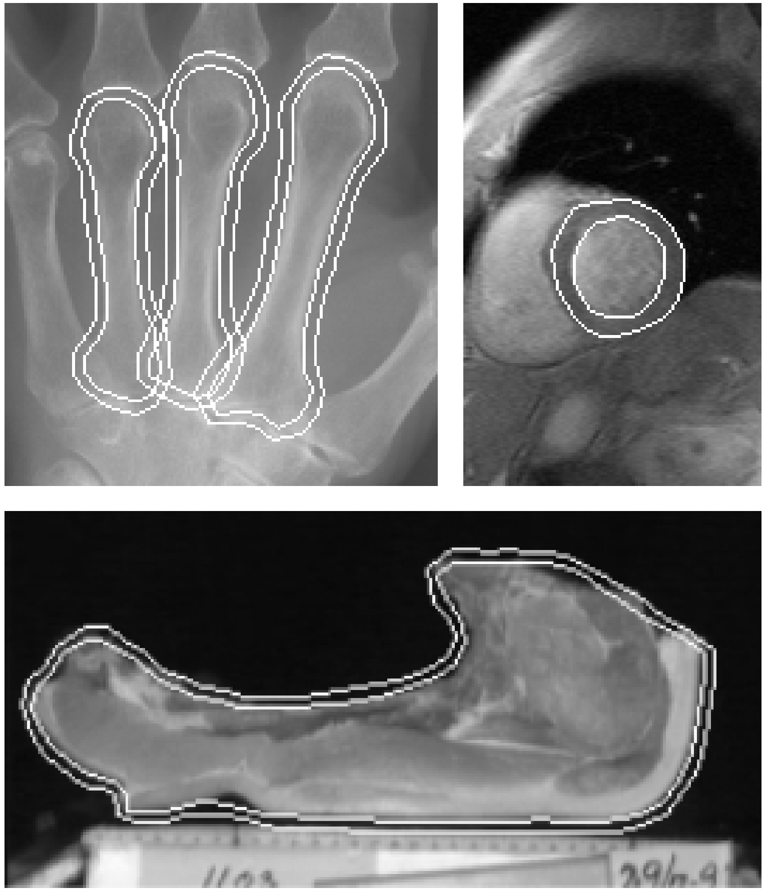


Figure D.6: Collage of the considered cases with successful segmentations. Upper left: A – Radiographs of Metacarpals. Upper right: B – Cardiovascular MRIs. Lower: C – Perspective images of Pork Chops. All images cropped to show details.

Common for all three cases is a relatively small training set (< 30). Using larger training sets the chances of over constraining the model can be reduced, leading to more (representative) flexibility. However, since the actual construction of the training sets is done by manually placing landmarks on the images – a tedious and error prone task – one wants to keep the training sets relatively small. Methods for (semi)-automatic extraction of landmarks are reviewed in [164].

Each experiment consisted of building an AAM leaving one image, $\boldsymbol{I}$, out. Subsequently the model was used to perform automatic segmentation in $\boldsymbol{I}$ using the proposed initialization scheme.[4] Each experiment was assed using the mean point to curve (point to associated border) measure and the mean intensity deviation measure [63]. Images were in grayscale 8-bit format. All figures in table D.1 are means over all leave-one-out experiments. Failure was declared when the point to curve error exceeded 10 pixels. Regarding performance the MRI-optimizations each took 200 ms on average on a PII 350 MHz.

The minimum and maximum error for the tests with lowest average error in each case was:

- A 3: min. 0.53 / max. 1.01 pixels (point to curve)
- B 5: min. 0.60 / max. 1.34 pixels (point to curve)
- C 3: min. 0.65 / max. 2.43 pixels (point to curve)

In case A, the adding of neighborhood made the model more specific, removing the single failure in A1. Further fine-tuning using simulated annealing increased not only the explicit optimization criteria, but also the landmark accuracy. The neighborhood adding in case B also yielded higher landmark accuracy. Due to large-scale

[4] Regarding the sparse sampling in the initialization phase, the x- and y-pose-component were sampled at a frequency of 12 pixels. $\boldsymbol{c}$-parameters were kept constant zero.

Table D.1: Leave-one-out test results for case studies.

#	Type	Point to curve deviation (pixels)	Mean intensity deviation	Init. fail.
A – Metacarpals				
1	Basic AAM	0.88	4.9	1
2	1+Neighborhood	0.84	5.2	0
3	2+SA	0.82	5.0	0
4	3+Lorentzian	0.83	5.0	0
B – Pork Chops				
1	Basic AAM	1.12	13.2	0
2	1+Neighborhood	0.91	13.9	0
3	2+SA	0.89	13.6	0
4	3+Lorentzian	0.91	13.6	0
5	Border AAM	0.86	23.5	0
C – Cardiac MRIs				
1	Basic AAM	1.18	7.1	0
2	1+Neighborhood	1.73	7.5	0
3	1+SA	1.06	5.9	0
4	3+Lorentzian	1.13	6.0	0

texture noise inside the object, the border AAM yielded the highest accuracy. Notice that the rather large texture error in B5 is not comparable to B1 – B4, since it is a completely different texture model. Contrary to case A, the cardiac AAM in case C possessed substantial structured contrast inside the object (the left ventricle), hence neighborhood-adding lead to a poorer fit. In all three cases, fine-tuning using simulated annealing improved both texture and landmark fit.

The lack of improvements using the robust Lorentzian similarity measure suggests that significant outliers as in fig. D.5 were not present.

D.5 Future work

We are currently investigating several methods, which extend the AAM scheme. Flexibility of the shapes are being enhanced by unifying Finite Element Models and AAMs by adding artificial interior points, which are deformed by an FEM. Further an Active Texture Weighting scheme is being designed which will add more flexibility to the texture model representation. Each texture pixel is given a weight, which is determined 1) manually, by drawing a semi-transparent mask or 2) automatically, by some function of the pixel variance over the training set. In the latter case, we expect this automatic method to supersede the manual decision of using a normal, neighborhood or border AAM. AAMs also extend to higher spatial dimensions [61], which will be the topic of our long-term future research.

D.6 Implementation

All experiments, illustrations etc. have been made using the Active Appearance Models Application Programmers Interface (AAM-API) developed in the C++ language. The API is released under the open source initiative, which means that others freely can download, use and elaborate on the AAM-API. Effort has been put into providing documentation and educational features such as movie generation of the modes of variation, model search etc. Further information on AAMs, source code and documentation can be obtained at *http://www.imm.dtu.dk/∼aam/*.

D.7 Conclusion

In this paper, we have presented a set of extensions which all yield higher landmark accuracy when applied to the type of situations they address. The proposed extensions are an enhanced representation of shape used for handling of homogeneous and heterogeneous textures and a fine-tuning of the AAM optimization with and without the usage of robust error norms. Finally, the usage of AAMs is fully automated by the presented initialization scheme.

The performance has been assessed on three different image modalities – i.e. radiographs, perspective images and magnetic resonance images, reaching a mean landmark location accuracy of 0.82, 0.86 and 1.06 pixels (pt.crv.), respectively, all using a relatively small training set of 23, 13 and 13 examples. The experiments were accomplished with no manual interaction. The implementation was unchanged in all three cases. No parameters were adjusted to produce the results.[5]

The three cases stress the fact that the AAM approach with the proposed extensions is a fully automated, general vision technique that captures domain knowledge through observation. Furthermore, we have experienced the AAM approach with the proposed extensions to be data-driven, self-contained and fast.

[5] To the benefit of future research activities and comparative studies, we intend to make all images and annotations available at our website.

D.8 Acknowledgements

Cardiac MRIs were provided and annotated by M.D. Jens Christian Nilsson and M.D. Bjrn A. Grnning, Danish Research Center of Magnetic Resonance, H:S Hvidovre Hospital. M.Sc.Eng. Torben Lund provided the annotation tool. M.D. Lars Hyldstrup, Department of Endocrinology, H:S Hvidovre Hospital provided, and Pronosco A/S digitized, the metacarpal radiographs. The work of M. B. Stegmann was partly supported by a grant from Pronosco A/S.

Appendix E

On parameter estimation in deformable models

Rune Fisker and Jens Michael Carstensen
Dep. of Mathematical Modelling, Tech. University of Denmark

Abstract

Deformable templates have been intensively studied in image analysis through the last decade, but despite its significance estimation of model parameters has received little attention. We present a method for supervised and unsupervised model parameter estimation using a general Bayesian formulation of deformable templates. In the supervised estimation the parameters are estimated using a likelihood and a least squares criterion given a training set. For most deformable template models the supervised estimation provides the opportunity for simulation of the prior model. The unsupervised method is based on a modified version of the EM algorithm. Experimental results for a deformable template used for textile inspection are presented.

E.1 Introduction

The general idea of deformable models is that a structure embedded in the image can be considered as a deformation of a given template. The deformable model is a Bayesian combination of two parts. One part which represents the prior knowledge about the structure, i.e. the deformable template, and a second part which represent the interactions with the observations (the image), the observation model. Deformable templates can roughly be separated into 2 groups: Free form and Parametric. Free form deformable templates have no explicit global structure because the prior only contains local continuity and smoothness constrains [33, 108, 118, 121]. In parametric deformable models prior knowledge of the global structure is included using a parameterized template of a specific structure [23, 64, 85, 102, 189].

Another characteristic of deformable models is that a number of model parameters, which gives the relative influence of different terms in the model, have to be selected. Though all deformable models contains model parameters the selection of these model parameters has received very little attention. [108, 189] don't comment on the selection and [23, 33, 64, 102] selects the model parameters based on empirical observations. [121] give guidelines for choosing the optimal parameters in the prior model based on bounds for the parameters. Only [118] use an unsupervised method based on a minimax criterion to determine the regularization parameter which gives the relative influence of the prior and observation part, respectively. We present a new method for supervised and unsupervised selection of all model parameters in a deformable model. This method also gives the opportunity for simulating the prior model for most deformable models. Section 2 contains a general Bayesian formulation of deformable models. Then the method for supervised and unsupervised model parameter estimation is presented, followed by experimental results and a conclusion.

E.2 Deformable models

A structure is modeled by a template that is uniquely described by a set of template parameters $\boldsymbol{v} = (v_1, v_2, ..., v_p)$. Let Ω_v denote the parameter space of $\boldsymbol{v}$. Using Bayes theorem deformable models can be described by the posterior probability $P(\boldsymbol{v}|\boldsymbol{y})$ of a realization of the template parameters $\boldsymbol{v}$ given an image $\boldsymbol{y}$. The posterior probability $P(\boldsymbol{v}|\boldsymbol{y})$ is defined as:

$$P(\boldsymbol{v}|\boldsymbol{y}) = \frac{P(\boldsymbol{v})P(\boldsymbol{y}|\boldsymbol{v})}{\sum_{\boldsymbol{v}\in\Omega_v} P(\boldsymbol{v})P(\boldsymbol{y}|\boldsymbol{v})} \tag{E.1}$$

where $P(\boldsymbol{v})$ is the prior model and $P(\boldsymbol{y}|\boldsymbol{v})$ is the observation model. [33, 108, 118, 121, 189] talks about energy and not about probabilities, but basically the formulations are analogous.

E.2.1 The prior model

The prior model represents the prior knowledge of the template parameter distribution. Typical the prior model consists of a number terms which represent different types of prior knowledge. Let the prior probability be Gibbs distributed and given by:

$$P(\boldsymbol{v}) \quad = \quad \frac{1}{Z_p(\boldsymbol{\alpha})} \exp\{-U(\boldsymbol{v}, \boldsymbol{\alpha})\} \tag{E.2}$$

where $Z_p(\boldsymbol{\alpha}) = \sum_{\boldsymbol{v}\in\Omega_v} \exp\{-U(\boldsymbol{v}, \boldsymbol{\alpha})\}$ is the normalizing constant, $U(\boldsymbol{v}) : \Omega_v \mapsto \boldsymbol{R}$ is the energy function representing the prior knowledge and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n) \in \Omega_\alpha$ are the prior parameters which gives the relative influence of the different terms.

Theorem 1: Let $U(\boldsymbol{v}, \boldsymbol{\alpha}) = \boldsymbol{v}'\boldsymbol{A}(\boldsymbol{\alpha})\boldsymbol{v} + \boldsymbol{b}(\boldsymbol{\alpha})\boldsymbol{v} + c(\boldsymbol{\alpha})$ depending only on $\boldsymbol{v}$ and $\boldsymbol{\alpha}$. If $\boldsymbol{A}(\boldsymbol{\alpha})$ is symmetric and positive definite and $\boldsymbol{b}(\boldsymbol{\alpha})$

belongs to the subspace defined by a linear mapping with $\boldsymbol{\Sigma}(\boldsymbol{\alpha})$ then $P(\boldsymbol{v}) \in N(\boldsymbol{\mu}(\boldsymbol{\alpha}), \boldsymbol{\Sigma}(\boldsymbol{\alpha}))$ and:

$$\begin{aligned} P(\boldsymbol{v}) = {} & \tfrac{1}{\sqrt{2\pi^p}} \tfrac{1}{\sqrt{det\boldsymbol{\Sigma}(\boldsymbol{\alpha})}} \times \\ & \exp\{-\tfrac{1}{2}(\boldsymbol{v} - \boldsymbol{\mu}(\boldsymbol{\alpha}))'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}(\boldsymbol{v} - \boldsymbol{\mu}(\boldsymbol{\alpha}))\} \end{aligned} \tag{E.3}$$

where $\mu(\boldsymbol{\alpha}) = -\boldsymbol{\Sigma}(\boldsymbol{\alpha})\boldsymbol{b}(\boldsymbol{\alpha})$ and $\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1} = 2\boldsymbol{A}(\boldsymbol{\alpha})$.

Proof: $U(\boldsymbol{v}, \boldsymbol{\alpha}) = \frac{1}{2}(\boldsymbol{v} - \boldsymbol{\mu}(\boldsymbol{\alpha}))'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}(\boldsymbol{v} - \boldsymbol{\mu}(\boldsymbol{\alpha})) = \frac{1}{2}\boldsymbol{v}'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}\boldsymbol{v} - \boldsymbol{\mu}'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}\boldsymbol{v} + \frac{1}{2}\boldsymbol{\mu}'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}\boldsymbol{\mu}$ i.e. $\mu(\boldsymbol{\alpha}) = -\boldsymbol{\Sigma}(\boldsymbol{\alpha})\boldsymbol{b}(\boldsymbol{\alpha})$ and $\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1} = 2\boldsymbol{A}(\boldsymbol{\alpha})$. □

If the prior (E.3) is rank deficient the conditional distributions $p(\boldsymbol{v} \setminus \boldsymbol{v}_s|\boldsymbol{v}_s)$ for $\boldsymbol{v}_s \subset \boldsymbol{v}$ will be rank sufficient. Theorem 1 implies that all $v_i \in \boldsymbol{v}$ has a prior distribution, i.e. $P(\boldsymbol{v})$ should be a function of all $v_i \in \boldsymbol{v}$. If this is not fulfilled then is the subset $\boldsymbol{v}_p \subset \boldsymbol{v}$, which have a prior distribution, used instead of $\boldsymbol{v}$. The assumption about $U(\boldsymbol{v}, \boldsymbol{\alpha})$ made in theorem 1 covers most deformable templates presented in the literature.

In the case where it can be justified that $U(\boldsymbol{v}, \boldsymbol{\alpha})$ fulfill the assumption made in theorem 1 the mean $\boldsymbol{\mu}(\boldsymbol{\alpha})$ can also be found as the solution to $\frac{\partial U(\boldsymbol{v}, \boldsymbol{\alpha})}{\partial \boldsymbol{v}} = 0$ and the inverse covariance $\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}$ as the Hessian $\frac{\partial^2 U(\boldsymbol{v}, \boldsymbol{\alpha})}{\partial^2 \boldsymbol{v}} = \boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}$. This method can be a fast alternative for determining $\boldsymbol{\mu}(\boldsymbol{\alpha})$ and $\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}$ if $U(\boldsymbol{v}, \boldsymbol{\alpha})$ is not directly in the form $\boldsymbol{v}'\boldsymbol{A}(\boldsymbol{\alpha})\boldsymbol{v} + \boldsymbol{b}(\boldsymbol{\alpha})\boldsymbol{v} + c(\boldsymbol{\alpha})$

E.2.2 The observation model

The observation model gives the probability for a given realization of $\boldsymbol{v}$ corresponds to the observations $\boldsymbol{y} \in \Omega_y$ - the image. In many cases the interaction corresponds to image intensity [23, 64, 85, 189] and/or edge information [33, 102, 108, 118, 121, 189] but in principle all kind of information can be combined e.g. texture or colour. Let the observation model be Gibbs distributed::

$$P(\boldsymbol{y}|\boldsymbol{v}) = \frac{1}{Z_o(\boldsymbol{\beta})} \exp\{-\frac{1}{\lambda} U(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\beta})\} \tag{E.4}$$

where $Z_o(\boldsymbol{\beta}) = \sum_{\boldsymbol{v} \in \Omega_v} \exp\{-\frac{1}{\lambda} U(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\beta})\}$ is the normalizing constant, $\lambda \in \Omega_\lambda$ is the regularization parameter that determines the relative influence of the prior and the observation model, $U(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\beta}) : \Omega_y \mapsto \boldsymbol{R}$ is the energy function representing the interaction between $\boldsymbol{y}$ and $\boldsymbol{v}$ and $\boldsymbol{\beta} = (\beta_1, ..., \beta_l) \in \Omega_\beta$ are the observation parameters which gives the relative influence of the different types of information.

E.3 Supervised model parameter estimation

Assume a training set of q template parameter sets $\boldsymbol{v}_1^t, .., \boldsymbol{v}_q^t$ corresponding to some structure within an image $\boldsymbol{y}$ are known.

E.3.1 The prior model

In the case where the assumption in theorem 1 is fulfilled the prior parameters can be estimated by using theorem 1 and the maximum likelihood (ML) estimator, which is defined as:

$$\hat{\boldsymbol{\alpha}} = \max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}, \boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t) \tag{E.5}$$

where it is assumed that $\hat{\boldsymbol{\alpha}} \in \Omega_\alpha$. If $\boldsymbol{v}_1^t, .., \boldsymbol{v}_q^t$ are stochastic independent and the covariance is rank sufficient then is the likelihood function directly given as:

$$L(\hat{\boldsymbol{\alpha}}, \boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t) = P(\boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t) = \prod_{k=1}^{q} P(\boldsymbol{v}_k^t) \tag{E.6}$$

where $P(\boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t)$ is the simultaneous density function for the prior distribution $P(\boldsymbol{v}_k^t)$ given by (E.3). If the covariance is rank deficient, i.e. $rank(\boldsymbol{\Sigma}) = p - r$ for $r > 0$, it is necessary to condition on at least r variables to obtain full rank. In this case e.g. the pseudo-likelihood [14] can be used as an approximation to the true likelihood:

$$\begin{aligned} PL(\hat{\boldsymbol{\alpha}}, \boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t) &= P(\boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t) \\ &= \prod_{k=1}^{q} \prod_{i=1}^{m} P(\boldsymbol{v}_{k,i}^t | \boldsymbol{v}_k^t \setminus \boldsymbol{v}_{k,i}^t) \end{aligned} \tag{E.7}$$

If $\boldsymbol{v}_1^t, .., \boldsymbol{v}_q^t$ are not stochastic independent then $P(\boldsymbol{v}_1^t, ..., \boldsymbol{v}_q^t)$ must be rewritten using conditional probabilities and Bayes theorem. When the prior parameters are estimated it is possible to simulate the prior model by sampling in the unconditional or conditional prior distribution.

E.3.2 The posterior model

Due to the image information in $P(\boldsymbol{y}|\boldsymbol{v})$ is it impossible to make any distribution assumption about $P(\boldsymbol{v}|\boldsymbol{y})$. This make it impossible to use the ML estimator for estimation of the observation and regularization parameters, because it is infeasible to calculate the normalizing constant even for small Ω_v. A likelihood approximation with the normalizing constant removed from $P(\boldsymbol{v}|\boldsymbol{y})$ or a criterion equivalent to the minimax criterion proposed by [118] can't be used either, because $P(\boldsymbol{v}|\boldsymbol{y})$ without the normalizing constant in many case would not be convex within $(\Omega_\lambda, \Omega_\beta)$. The observation and regularization parameters are instead estimated as the parameters which minimize the least squares error (LSE) between the training set parameters $\boldsymbol{v}_1^t, .., \boldsymbol{v}_q^t$ and the estimated parameters $\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q$:

$$(\hat{\lambda}, \hat{\boldsymbol{\beta}}) = \min_{\lambda, \boldsymbol{\beta}} \frac{1}{pq} \sum_{i=1}^{q} (\boldsymbol{v}_i^t - \hat{\boldsymbol{v}}_i)' \boldsymbol{C} (\boldsymbol{v}_i^t - \hat{\boldsymbol{v}}_i) \tag{E.8}$$

where $\boldsymbol{C}$ is diagonal matrix where each parameter can be assigned an estimation weight (in most cases $\boldsymbol{C}$ is chosen equal to the identity matrix $\boldsymbol{I}$) and $\hat{\boldsymbol{v}}_i$ is the maximum a posteriori (MAP) estimate of parameter set i defined as $\hat{\boldsymbol{v}}_i = \max_{\boldsymbol{v}_i} P(\boldsymbol{v}_i|\boldsymbol{y})$. Many different technics have been applied for MAP estimation as deterministic [33, 102, 108, 118, 189], stochastic [23, 85] and heuristic optimization algorithms [64].

If the assumption in theorem 1 is not fulfilled the model parameters $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$ can still be estimated by expanding (E.8) to:

$$(\hat{\boldsymbol{\alpha}}, \hat{\lambda}, \hat{\boldsymbol{\beta}}) = \min_{\boldsymbol{\alpha},\lambda,\boldsymbol{\beta}} \frac{1}{pq} \sum_{i=1}^{q} (\boldsymbol{v}_i^t - \hat{\boldsymbol{v}}_i)' \boldsymbol{C} (\boldsymbol{v}_i^t - \hat{\boldsymbol{v}}_i) \tag{E.9}$$

If $log\ P(\boldsymbol{v}|\boldsymbol{y})$ is a linear function of $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$ then (E.9) have to be solved with respect to the constraint: $\lambda + \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{l} \beta_i = constant$ to limit the number of solutions to one.

E.4 Unsupervised model parameter estimation

The unsupervised model parameter estimation is based on a modified version of the Expectation-Maximization (EM) algorithm [51]:

1. Start with the observations $\boldsymbol{y}$, an initial estimate of $\boldsymbol{v}$ and a guess for $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$.
2. Estimate $\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y})$ using the current $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$.
3. Use the algorithm for supervised model parameter estimation (Section E.3) to estimate $(\hat{\boldsymbol{\alpha}}, \hat{\lambda}, \hat{\boldsymbol{\beta}})$ taking $\hat{\boldsymbol{v}}$ as training set.
4. Go to 2 for a number of iterations or until $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$ has approximately converged.

E.5 Experimental results

A deformable template is used for localization of the horizontal yarns in a system for automated visual inspection of textile [64]. The system should be used for inspection of above 35 different types of textile and new types are developed frequently, so it is necessary that a non-expert can train the system, i.e. select the model parameters $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$.

The horizontal yarn k is modelled as $\boldsymbol{v}_k = (v_{k,1}, v_{k,1}, v_{k,2}, ..., v_{k,p})'$, where $v_{k,j}$ is the vertical position of yarn k in the vertical space j between the vertical yarns. The horizontal position of the vertical space j is assumed known, because it is easily located as a local maxima in a vertical projection of the image [64]. The distance between each vertical space j is assumed to be constant.

The posterior probability of the deformable model used for textile inspection is given by [64]:

$$\begin{aligned} &P(\boldsymbol{v}_k|\boldsymbol{v}_{k-1}, \boldsymbol{y}) = \frac{1}{Z_o(\boldsymbol{\alpha}) Z_p(\boldsymbol{\alpha})} \times \\ &\exp\{-\alpha_1 \textstyle\sum_{j=2}^{p} (v_{k,j} - v_{k,j-1})^2 \\ &-\alpha_2 \textstyle\sum_{j=1}^{p} (v_{k,j} - v_{k-1,j} - d)^2 \\ &-\tfrac{1}{\lambda} \textstyle\sum_{j=1}^{p} I(v_{k,j}, j)\} \end{aligned} \tag{E.10}$$

where the two first terms, $U(\boldsymbol{v}, \boldsymbol{\alpha}) = \alpha_1 \sum_{j=2}^{p} (v_{k,j} - v_{k,j-1})^2 + \alpha_2 \sum_{j=1}^{p} (v_{k,j} - v_{k-1,j} - d)^2$, correspond to the prior model. The first term favors strictly horizontal threads. The second term describes that the thread $\boldsymbol{v}_k$ should be placed in a predefined distance $d \in \boldsymbol{R}_+$ from $\boldsymbol{v}_{k-1}$. The third term in $P(\boldsymbol{v}_k|\boldsymbol{v}_{k-1}, \boldsymbol{y})$ is the observation model where $I(v_{k,j}, j)$ is the negative horizontal mean at the vertical position $v_{k,j}$ of the pixels in the vertical space j. $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) \in \boldsymbol{R}_+^2$ are the prior parameters and $\lambda \in \boldsymbol{R}_+$ is the regularization parameter. Let $U(\boldsymbol{v}, \boldsymbol{\alpha}) = \alpha_1 (\boldsymbol{N}\boldsymbol{v}_k)'(\boldsymbol{N}\boldsymbol{v}_k) + \alpha_2 (\boldsymbol{v}_k - (\boldsymbol{v}_{k-1} + \boldsymbol{d}))'(\boldsymbol{v}_k - (\boldsymbol{v}_{k-1} + \boldsymbol{d})) = \alpha_1 \boldsymbol{v}_k' \boldsymbol{N}' \boldsymbol{N} \boldsymbol{v}_k + \alpha_2 (\boldsymbol{v}_k \boldsymbol{I} \boldsymbol{v}_k - 2(\boldsymbol{v}_{k-1} + \boldsymbol{d}) \boldsymbol{v}_k + (\boldsymbol{v}_{k-1} + \boldsymbol{d})'(\boldsymbol{v}_{k-1} + \boldsymbol{d}) = \boldsymbol{v}_k' (\alpha_1 \boldsymbol{N}' \boldsymbol{N} + \alpha_2 \boldsymbol{I}) \boldsymbol{v}_k - 2\alpha_2 (\boldsymbol{v}_{k-1} + \boldsymbol{d}) \boldsymbol{v}_k + (\boldsymbol{v}_{k-1} + \boldsymbol{d})'(\boldsymbol{v}_{k-1} + \boldsymbol{d})$

where

$$N = \begin{bmatrix} 0 & 0 & \dots & 0 \\ -1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{E.11}$$

$$N'N = \begin{bmatrix} 1 & -1 & \dots & 0 \\ -1 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{E.12}$$

From above it's seen that the prior model fulfill the assumption in theorem 1 (which was obvious because the prior only consists of quadratic terms). By theorem 1 $P(\boldsymbol{v}_k|\boldsymbol{v}_{k-1}) \in N(\boldsymbol{\mu}_k(\boldsymbol{\alpha}), \boldsymbol{\Sigma}(\boldsymbol{\alpha}))$ where:

$$\begin{aligned} &\boldsymbol{\mu}_k(\boldsymbol{\alpha}) = \boldsymbol{\Sigma}(\boldsymbol{\alpha})2\alpha_2(\boldsymbol{v}_{k-1} + \boldsymbol{d}) \\ &\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1} = \\ &\begin{bmatrix} 2\alpha_1 + 2\alpha_2 & -2\alpha_1 & \dots & 0 \\ -2\alpha_1 & 4\alpha_1 + 2\alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 2\alpha_1 + 2\alpha_2 \end{bmatrix} \end{aligned} \tag{E.13}$$

E.5.1 Estimation of model parameters

Assume a training set of 150 yarns $\boldsymbol{v}_1^t, ..., \boldsymbol{v}_{150}^t$ is manual marked in a image by an operator, see subset in figure E.2 and E.1. It can be shown that the rank of $\boldsymbol{\Sigma}^{-1}$ is full $\forall(\alpha_1, \alpha_2) \in \boldsymbol{R}_+^2$, but the prior probabilities are not independent. Because the prior probability is a Markov Random field and by using Bayes theorem the likelihood is given by:

$$\begin{aligned} &L(\boldsymbol{\alpha}, \boldsymbol{v}_1^t, ..., \boldsymbol{v}_{150}^t) \\ &= P(\boldsymbol{v}_1^t, ..., \boldsymbol{v}_{150}^t) \\ &= P(\boldsymbol{v}_1^t)P(\boldsymbol{v}_2^t, ..., \boldsymbol{v}_{150}^t|\boldsymbol{v}_1^t) \\ &= P(\boldsymbol{v}_1^t)P(\boldsymbol{v}_2^t|\boldsymbol{v}_1^t)P(\boldsymbol{v}_3^t, ..., \boldsymbol{v}_{150}^t|\boldsymbol{v}_2^t) \\ &= P(\boldsymbol{v}_1^t)\prod_{k=2}^{150} P(\boldsymbol{v}_k^t|\boldsymbol{v}_{k-1}^t) \end{aligned} \tag{E.14}$$

where $P(\boldsymbol{v}_k^t|\boldsymbol{v}_{k-1}^t) = \frac{1}{\sqrt{2\pi}^p}\frac{1}{\sqrt{det\boldsymbol{\Sigma}(\boldsymbol{\alpha})}}\exp\{-0.5(\boldsymbol{v}_k^t - \boldsymbol{\Sigma}(\boldsymbol{\alpha})2\alpha_2(\boldsymbol{v}_{k-1}^t + \boldsymbol{D}))'\boldsymbol{\Sigma}(\boldsymbol{\alpha})^{-1}(\boldsymbol{v}_k^t - \boldsymbol{\Sigma}(\boldsymbol{\alpha})2\alpha_2(\boldsymbol{v}_{k-1}^t + \boldsymbol{D}))\}$ and $P(\boldsymbol{v}_1^t)$ is constant because $\boldsymbol{v}_1$ is estimated using an ad hoc procedure.

Prior parameters $\alpha_1 = 1.0476$ and $\alpha_2 = 0.0172$ are then estimated for the known yarns corresponding to figure E.1. Simulations of the prior model are then performed by sampling in the Gaussian distribution $N(\boldsymbol{\mu}_k(\boldsymbol{\alpha}), \boldsymbol{\Sigma}(\boldsymbol{\alpha}))$, see figure E.1. If the simulated horizontal yarns are compared to the real yarns this seems to verify that the prior model with the estimated parameters is a good model of the yarns. The regularization parameter $\lambda = 0.3652$ is then estimated using the LSE (E.8) with $\boldsymbol{C} = \boldsymbol{I}$ and a heuristic MAP-estimation algorithm [64]. The yarns in the image in figure E.2 is located using the estimated parameters with very good results, see figure E.2. To examine the variation of the estimated parameters within the same textile sample, two parameter sets have been estimated on two different pieces of one sample. This was done for two different samples and the mean variation on α_1, α_2 and λ was 1.9%, 5.2% and 9.5%, respectively. The parameter variation between samples depends on the amount of variations and defects within the samples used for estimation. This is so significant that the estimated parameters often can be used to discriminate between god and bad samples. Using 5 representative samples were the parameters estimated with the mean $\alpha_1 = 1.0214, \alpha_2 = 0.0202$ and $\lambda = 0.4145$ and the standard deviation 0.2009, 0.0066 and 0.0717, respectively. The parameters were then used to locate the yarns in the 5 samples with very good results leading to an average increase on only 6% in the LSE (E.8) compared to the LSE obtained with the individual estimated parameters. The method for supervised parameter estimation have also been tested on other types of textile with good results.

The unsupervised method performs well, but it is a little bit sensitive to the initial guess $(\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})$ because it tends to get caught in local maxima if the regularization parameter is chosen very large or small. This sensitivity is a well known problem for the EM algorithm. For 6 randomly chosen $(\boldsymbol{\alpha}, \frac{1}{\lambda}, \boldsymbol{\beta}) \in [0.2; 2]$ under the constrain $\alpha_1 < \frac{1}{\lambda}$ and $\alpha_2 < \frac{1}{\lambda}$ were the parameters corresponding to the yarns in figure E.2 estimated with the mean $\alpha_1 = 1.3429, \alpha_2 = 0.0059$ and $\lambda = 0.5050$ and the standard deviation 0.1708, 0.0001 and 0.1005. The algorithm converged within 10 iterations. If the estimated parameters are compared with the previous estimated parameter there exists some differences, but still the LSE is only increased by 14 % compared to the LSE for the individual estimated parameters, and the yarns are still located very well, see figure E.2. Similar results for the unsupervised parameter estimation are obtained for other textile samples and types.

E.6 Conclusion

A method for supervised and unsupervised estimation of model parameters in deformable templates have been presented. Experimental results are successful and indicates that the methods are robust, though the unsupervised method is a little sensitive to the initial parameter configurations. The opportunity for simulations of the prior model seems to be a good tool for verification of the model. The presented methods also contains interesting perspectives regarding using the estimated parameters as features for description of the located structure and regarding automated model selection using an information criterion.

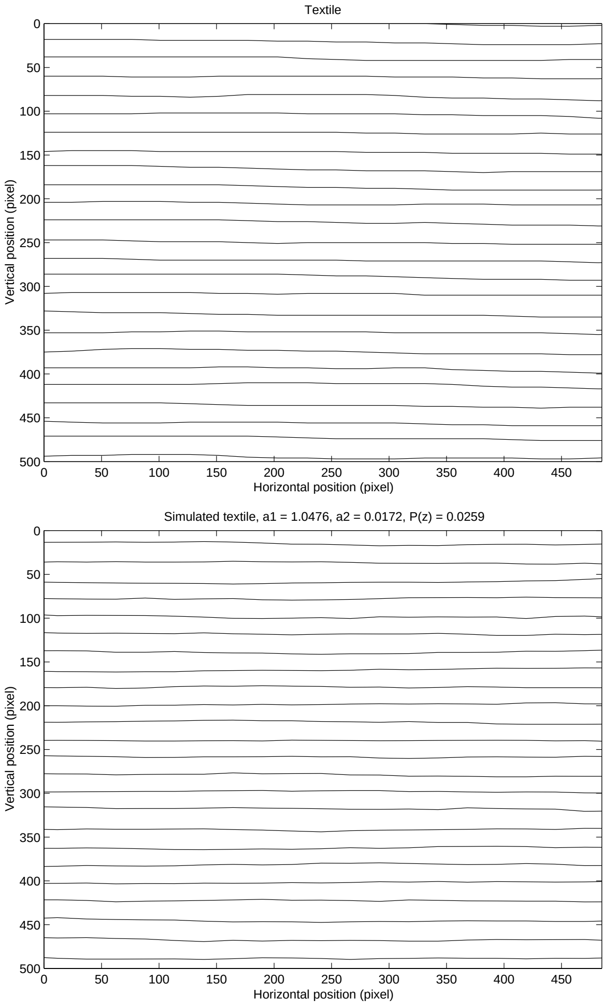


Figure E.1: Manually marked horizontal yarns (top) and simulated horizontal yarns (bottom)

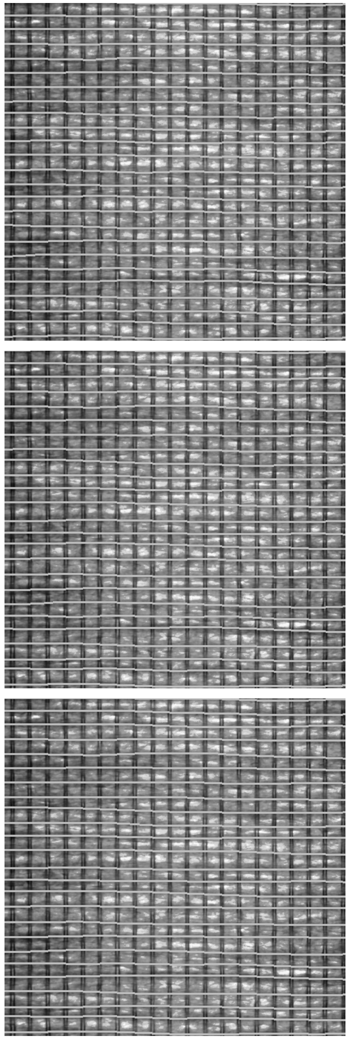


Figure E.2: Textile with manually marked horizontal yarns (top), yarns located using supervised estimated parameters (center) and yarns located using unsupervised estimated parameters (bottom).

Appendix F

Initialization and Optimization of Deformable Models

Rune Fisker, Jens Michael Carstensen and Kaj Madsen
Dep. of Mathematical Modelling, Tech. University of Denmark

Abstract

The deformable model literature has in general been very focused on the formulation and development of new models or the solution of a specific application. The final and crucial steps of initialization and optimization of the deformable model, needed for making inference, have received very little attention. We present an improved general method for initialization based on a filter interpretation of the likelihood. This method will, for a large number of models and problems, generate the initialization significantly faster than the previously proposed general methods. To perform the optimization a number of general algorithms from the optimization literature is introduced, which are novel in the context of deformable models. These algorithms are compared to general algorithms previously used for optimization of deformable models. Tests are performed on real data.

Index terms: Deformable models, active contour models, deformable templates, initialization and optimization.

F.1 Introduction

The group of models known as deformable models have been presented under many different names, where the best known probably are active contour models (snakes), active shape models and deformable templates. One of the characteristics of deformable models is that an object or structure embedded in an image is represented by a vector of template parameters $\boldsymbol{v} \in \Omega_v$, where Ω_v is the parameter space. Direct or under some mapping the template parameters $\boldsymbol{v}$ defines the structure. To make inference about the object in an image $\boldsymbol{y}$ a probabilistic distribution $P(\boldsymbol{v}|\boldsymbol{y})$ (or an equivalent energy function) is formulated such that the configuration of the template parameters, which best match the object, have the highest probability. The probabilistic distribution can be separated in two parts: the prior $P(\boldsymbol{v})$ and the likelihood $P(\boldsymbol{y}|\boldsymbol{v})$. The prior distribution $P(\boldsymbol{v})$ represents the prior knowledge about the structure and is independent of the image, and the likelihood $P(\boldsymbol{y}|\boldsymbol{v})$ (or the observation model) represents the interactions with the image (the observations). Using Bayes theorem the prior and likelihood is combined to the final posterior distribution used to make interference:

$$P(\boldsymbol{v}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{v})P(\boldsymbol{v}) \tag{F.1}$$

Some authors use an energy formulation of deformable models and not a probabilistic or Bayesian one, but basically the formulations are analogous [65]. In general we prefer a Bayesian formulation, because it gives a natural separation of model and image contributions in $P(\boldsymbol{v})$ and $P(\boldsymbol{y}|\boldsymbol{v})$, and it provides the opportunity to simulate and thus visualize the appropriateness of the prior model. Further it provides several different ways to make inference in $P(\boldsymbol{v}|\boldsymbol{y})$.

F.2 Deformable models

Jain et al. [102] divide deformable models into two groups: Free form and Parametric. Free form deformable templates have no explicit global structure because the prior only contains local continuity and smoothness constraints. This makes free form templates able to represent an arbitrary shape as long as the continuity and smoothness constraints are satisfied. The best known example of free form models is active contour models or snakes originally proposed by Kass et al. [108]. To compensate for the shrinking effect of closed active contour models Cohen and Cohen [33, 34] add an inflation force to the snake model. Another example is Lai and Chin [118], who propose a more generalized formulation of active contour models.

In parametric deformable models prior knowledge of the global structure is included using a parameterized template. One of the pioneers of deformable models is Grenander, who proposes a parametric model [85] based on a 2D vector cycle representation of the given object and a stochastic distribution that governs the deformations of the initial template. This model has been further developed and described in [86, 155]. Jain et al. [102] propose a general framework for object matching based on prototype (or average) templates and with a different type of deformations and local constrains on these deformations. A similar model is proposed by Garrido and Blanca [72], but they use different methods for initialization and optimization. Another general stochastic model is proposed by Staib et al. [163]. This model uses elliptic Fourier descriptors to represent the boundary of the object. Cootes et al. [41] propose an active shape

model, where the object is represented by the mean shape of a training set and a linear combination of eigenvectors of the shape variation from this mean. This work has further been developed into the novel Active Appearance models [35], which also incorporates the appearance, i.e. grey level information. All the previous cited parametric models are general, because they, given a prototype template or a training set, are able to handle an almost arbitrary object. Probably the largest group of deformable models are formulated and tuned for a specific object, because it is often needed to incorporate special assumption to be able to make inference about a specific object. The properly best known model formulated for a specific problem is an eye model and a mouth model proposed by Yuille et al. [189]. A description of more models can be found in the excellent book on Active Contours by Blake and Isard [18] and in the survey papers by McInerney and Terzopoulos [132] and by Jain et al. [101].

Most of the deformable models presented in the literature have a prior model $P(\boldsymbol{v})$, which penalize the deviation of a given parameter configuration with respect to the prototype/ideal configuration. In most cases, the prior model is assumed to be Gibbs distributed, which gives the opportunity to simulate the model by sampling in the conditional or unconditional prior distribution [23, 65]. In many cases the likelihood model $P(\boldsymbol{y}|\boldsymbol{v})$ is formulated such that the interaction between a realization of the template $\boldsymbol{v}$ and an observed image $\boldsymbol{y}$ corresponds to image intensity [23, 41, 35, 85, 86, 167, 155, 189] and/or gradient (edge) information [33, 34, 72, 101, 102, 108, 118, 163, 189], but in principle all kinds of information can be combined e.g. texture or colour. Another characteristic of most deformable models is that a number of model weight parameters, which gives the relative influence of different terms in the model, have to be selected. Most authors select the weight parameters based on empirical observations. A method for supervised and unsupervised estimation of the parameters is proposed by Fisker and Carstensen [65].

F.3 Initialization

The formulated or chosen deformable model is then used to make inference about a structure in an image, e.g. by estimating the template parameters, which corresponds to the maximum a posteriori (MAP), defined as:

$$\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y}) \tag{F.2}$$

The MAP estimation is performed using an iterative optimization procedure, which presume an initial configuration $\boldsymbol{v}^0$ of the template parameters. This basically separates the MAP estimation in two steps: initialization and optimization.

The request for the initial configuration is, that it should be relatively close to the real object in the image to achieve successful refinement during the optimization. A number of authors initialize the template manually [33, 34, 108, 189], use ad hoc methods for their specific problem [23, 85, 155] or do not comment on the issue [35, 167]. Only a few authors propose more general methods for initializations. Staib et al. [163] use the mode of the prior distribution for initialization, i.e. the configuration of the template parameters which have the highest prior probability. Unfortunately this will only work, when there exist a non-uniform prior for all template parameters, which is not the case in general. Almost all models are for example invariant to translation, i.e. a uniform prior on translation. Lai and Chin [118] and Garrido and Blanca [72] propose to use the generalized Hough transform [11] and a modified version of the generalized Hough transform, respectively. The algorithms presented only work for models using a gradient based likelihood model and can not be used in the general case. Jain et al. [102] propose a search strategy, where the general idea is to search for the configurations in parameter space with the highest posterior probabilities. The configurations with a probability above some threshold are then used as the initial configurations. To improve the computational efficiency of the search a sub-sampled template is used. Jain et al. [102] also smooth the likelihood. The general idea of using modes of the posterior density, i.e. configurations corresponding to local maxima, as initial configurations means that this approach can be applied to any deformable model. Another advantage is that the function which calculate the likelihood and prior probability as a function of the template parameters $\boldsymbol{v}$ during the optimization directly can be used for initialization as well. This way of calculating the likelihood is defined as direct calculation.

In principle, the search for maxima should be performed for variations of all parameters in the template, but in practice the search should only be performed on a relevant and feasible set of parameters. Jain et al. [102] for example only search evenly spaced positions combined with a discrete set of orientations. Although the search is performed on a limited number of different parameter sets the computational cost is often huge.

In the typical case the search only need to be performed on the position occasionally combined with a search at different rotations and scales. The rest of the template parameters are chosen such that they optimize the prior distribution. In this case, the prior distribution is often constant for a large number of configurations, because most models are invariant to translation and rotation. This means that the computational cost only come from the likelihood calculations. On this basis we propose a method for calculating the likelihood for a large number of different positions, which significantly reduces the computational cost compared to the cost for the direct calculations. The fundamental idea is that for a given configuration of the template, the likelihood function can be interpreted as a rigid filter, which is convolved with the relevant images at a given position. This approach presumes that the likelihood can be interpreted as a filter, but this is almost always the case. In the normal case where the template change all the time this is without interest, but this is not the case when the template is constant for a large number of likelihood evaluations at different positions. The calculation of the likelihood for a given configuration of the template for a large number of positions can then simply be done by a convolution of the relevant image with the corresponding filter. As shown under the experimental results this approach can lead to a significant reduction of the computational cost for initialization. The gain in computation time depends of course on how sparse the positions are sampled and on the used deformable model and image. In general, the advantage increases when the difference between the size of the image and the template increases and when the distance between each sampled position decreases. The filters can be computed before the initialization, so this will not increase the computational load. Problems can arise along the borders of the image depending on how the border properties of the convolution algorithm and the deformable model are defined. Direct calculating of the likelihood probability along the borders can of course solve this problem, but in most cases this is not a problem, because the entire structure is assumed to be inside the image. In the case of more than one combination of template parameters, e.g. parameters corresponding to a number of different scales, a corresponding bank of likelihood interpreted filter can be used.

F.4 Optimization

The final optimization is typically a medium-to-high-dimensional optimization problem. The problem can, in most cases, be categorized as continuous unconstrained optimization of a nonlinear function. The problem is continuous because the template parameters are real, i.e. $\Omega_v \subset \mathbb{R}^N$ where N is the number of template parameters, and unconstrained because most authors do not work with hard constrains. In reality there exists constrains, e.g. due to the limited size of the images, but in practice this is not a problem for most models, which is partly because the prior distribution acts as a penalty function, that forces the solution to stay inside or close to the feasible region. Note that maximizing the probability according to (F.2) is equivalent to minimizing the energy of a deformable model, because the energy

$U(\boldsymbol{v}, \boldsymbol{y}) = -log(P(\boldsymbol{v}|\boldsymbol{y})) + constant$ under the weak assumption that $P(\boldsymbol{v}|\boldsymbol{y})$ is Gibbs distributed. To avoid confusion, the optimization will only deal with the energy minimization perspective.

A number of different optimization methods have been applied to perform the optimization. In [23, 85, 155, 167] simulated annealing [27, 112] is used for optimization. Another popular method is the steepest descent approached used in [33, 72, 102, 101, 108, 118, 163, 189]. Other approaches for the optimization are dynamic programming [4] and different heuristics [35, 41]. Dynamic programming can not be used in the general case, because it presumes a logical order of the parameters. Likewise it is not possible to use the heuristics in the general case, because they are tuned to a specific problem and/or model. The computational cost and robustness of the optimization can be improved by using a multiresolution strategy [42, 102].

When the generally applied methods for optimization of deformable models are compared to the literature for continuous unconstrained optimization the difference is significant. The optimization literature is very focused on the more advanced gradient based methods and the disadvantages of steepest descent. Another important issue in the optimization literature, which has received very little attention in the deformable model literature, is the speed of convergence of the different methods. These reflections are the main motivation for comparing the following optimization methods:

- *Pattern Search (PS) [94].*
- *Steepest Descent (SD) .*
- *Conjugate Gradient with Fletcher-Reeves update and without resetting (CG).*
- *Quasi-Newton (BFGS).*
- *Simulated Annealing (SA) [27, 112].*

Pattern search is a non-gradient based optimization algorithm, which has received increasing interest in the optimization community. Conjugate gradient and BFGS are well known optimization methods (see e.g [52]), which should have better properties than steepest descent.

The gradient based methods create the problem of estimating the gradient $\boldsymbol{g}(\boldsymbol{v}) = (g_1(\boldsymbol{v}), ..., g_N(\boldsymbol{v}))^T = (\frac{\partial U(\boldsymbol{v},\boldsymbol{y})}{\partial v_1}, ..., \frac{\partial U(\boldsymbol{v},\boldsymbol{y})}{\partial v_N})$, which in general is impossible for the likelihood by analytic means. In practice the gradient can be estimated by finite differences $g_i(\boldsymbol{v}) \approx (U(\boldsymbol{v}+h_i\boldsymbol{e}_i, \boldsymbol{y}) - U(\boldsymbol{v}, \boldsymbol{y}))/h_i$ (forward differences) or $g_i(\boldsymbol{v}) \approx (U(\boldsymbol{v}+h_i\boldsymbol{e}_i, \boldsymbol{y}) - U(\boldsymbol{v}-h_i\boldsymbol{e}_i, \boldsymbol{y}))/(2h_i)$ (central differences), where $\boldsymbol{e}_i$ and h_i are the unit vector and the step size in the i'th direction, respectively. A third option is to make a linear fit to a number of energies $U(\boldsymbol{v}+h_1\boldsymbol{e}_i, \boldsymbol{y}), ..., U(\boldsymbol{v}+h_M\boldsymbol{e}_i, \boldsymbol{y})$ to get an even more robust estimate of the gradient.

Another issue for the gradient based methods is, when a descent direction s has been determined, to decide how long the step γ_s should be in this direction. The search along the descent direction is known as line search. In the early days of optimization the exact line search was dominating, but today soft line searches (see e.g [52]) are gaining in popularity. Soft line searches approximate the step γ_s corresponding to the minimum along s, by terminating the search under some weak criterion. Soft line search should in general be faster than exact line search.

For simulated annealing it is necessary to choose a temperature scheme and a method for generating a new configuration. The best way to generate a new random configuration is by sampling in the posterior distribution or almost as good by sampling in the prior distribution like [85, 155, 167]. Unfortunately this is not possible in the general case, because the distributions often are improper. On this basis a Monte Carlo random walk method is chosen, where independent white noise is added to each template parameter. An exponential temperature scheme is chosen, i.e. $T^{t+1} = kT^t$.

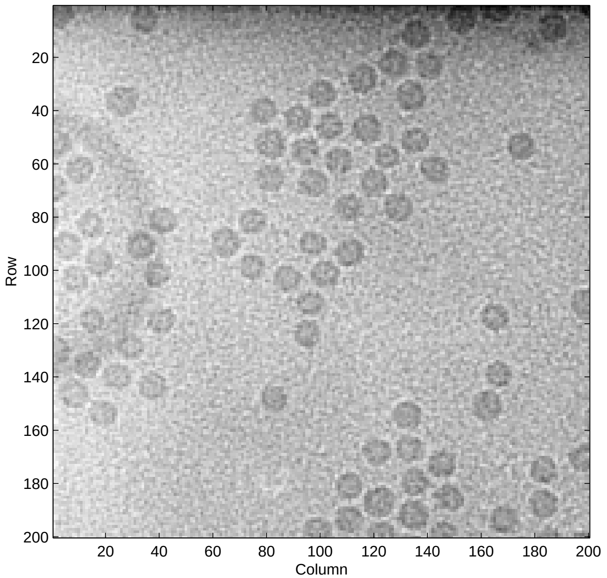


Figure F.1: Original close-up image y_o of nanoparticles.

F.5 Experimental results

The methods for initialization and optimization have been tested on a simple deformable model used to estimate the size and shape characteristics of nanoparticles in scanning electron microscope images, see figure F.1. These nanoparticles are known to be approximately elliptical. To impose the elliptical shape on the particles is used a deformable ellipse model. The parameters of the located ellipses capture the relevant size and shape characteristics. The presence of significant noise makes more simple approaches like direct filtering end up with highly irregular objects representing the particles. A particle is modelled by an ellipse with the template parameters:

$$\boldsymbol{v} = (r_0, c_0, a, b, \phi) \tag{F.3}$$

where (r_0, c_0) is the ellipse centre, a and b are the ellipse axes and ϕ is the rotation relative to horizontal. The a posteriori probability is given by:

$$\begin{aligned} P(\boldsymbol{v}|\boldsymbol{y}) = \tfrac{1}{Z}exp\{-(1-\tfrac{a}{b})^2 - \tfrac{1}{\lambda}(\tfrac{1}{|\Omega_{in}|}\textstyle\sum_{(r,c)\in\Omega_{in}} y(r,c) \\ -\tfrac{1}{|\Omega_{out}|}\textstyle\sum_{(r,c)\in\Omega_{out}} y(r,c))\} \end{aligned} \tag{F.4}$$

where $\Omega_{in} = \{(r,c)|\frac{(r_r)^2}{a^2} + \frac{(c_r)^2}{b^2} \leq 1\}$ and $\Omega_{out} = \{(r,c)|1 < \frac{(r_r)^2}{a^2} + \frac{(c_r)^2}{b^2} \leq 2\}$ where $r_r = (r - r_0)cos(\phi) + (c - c_0)sin(\phi)$ and $c_r = (c - c_0)cos(\phi) - (r - r_0)sin(\phi)$, $y(r,c)$ is the image intensity at row r and column c, λ is the regularization constant and Z is the normalizing constant. The first term corresponds to the prior model $P(\boldsymbol{v})$ and the next two to the likelihood $P(\boldsymbol{y}|\boldsymbol{v})$. The interpretation of the prior model is that the ideal particle is circular. The prior model is invariant to translation, rotation, and scale, which also seems to be a fair assumption. The interpretation of the likelihood is that the mean of the pixels in a band around the ellipse are subtracted from the mean of the pixels within the ellipse, where the band have the same area as the area of the ellipse. The probability is then increased by maximizing the difference. The regularization constant determines the relative influence of the prior and likelihood. Under normal circumstances the regularization constant λ should be estimated according to [65], but to make it possible to compare the energy for the different optimization methods λ have to be the same, so λ is set to $\lambda = 3$.

F.5.1 Initialization

The first step to use the proposed method for initialization is to make a filter interpretation of the likelihood function. It is assumed that an approximated radius r is known. The axis of the ellipse is then set to $a = b = r$ to maximize the prior probability. With $a = b = r$ the model is rotation invariant and an approximate scale is known, so only the different positions need to be evaluated. To be able to

locate the ellipses the likelihood needs to be calculated with a sample distance of one pixel in the vertical and horizontal positions in the image at full resolution. This is equivalent to calculating the likelihood for the centre of the ellipse (r_0, c_0) equal to each pixel position. Based on these assumptions a filter $f_L(a, b, \phi)$ equivalent to the likelihood can be derived. Given a radius r it is straight forward to determine the pixels which belongs to the inside of the ellipse and to the band around the ellipse according to (F.4). An example of the likelihood interpreted filter $f_L(5, 5, 0)$ corresponding to $a = b = r = 5$ can be seen in figure F.2. The likelihood $P(\boldsymbol{y}|r_c, c_c, a, b, \phi)$ is then calculated by a convolution of the filter $f_L(a, b, \phi)$ and the image $\boldsymbol{y}$. The likelihood corresponding to $P(\boldsymbol{y}|r_c, c_c, a, b, \phi)$ is then identical to the pixel value (r_c, c_c) in the filtered image $L_{(a,b,\phi)}$. The result of a convolution of the filter $f_L(5, 5, 0)$ corresponding to $r = 5$ and the image in figure F.1 can be seen in figure F.3. It is assumed, that the entire ellipse and the surrounding band should be inside the image, such that all pixels closer to the border than $\sqrt{2}r$ are set equal to zero. A comparison of the computation time for the direct calculations according to (F.4) and the filter calculations of the likelihood in

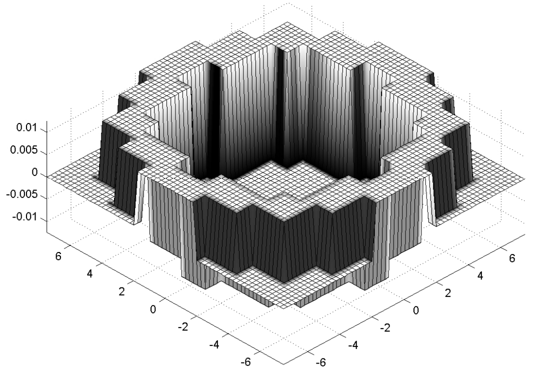


Figure F.2: Likelihood interpreted filter $f_L(5, 5, 0)$ corresponding to $r = 5$.

figure F.3 shows, that the filter approach is approximately 550 times faster in a C++ implementation and 1400 times faster in an identical Matlab implementation. These results of course depends on the actual implementation of the direct calculations and the convolution. The direct calculations are performed by independent repeated call to an optimized function, which calculate the likelihood corresponding to the template parameters $\boldsymbol{v}$. The speed of the direct calculation could of course be increased, if the implementation used the knowledge, that some parameters are constant for a large number of calculations, but then in most cases it would be impossible to use the same function for likelihood calculation during the optimization. To create the initial configurations the likelihood image is thresholded and labelled, see figure F.4. Each blob is assumed to correspond to a nanoparticle. The initial configuration for each ellipse model is then generated by setting the centre of the ellipse equal to the centre of gravity of the corresponding blob, the main axis $a = b = r$ and the orientation $\phi = 0$. The initialization procedure locate all the nanoparticles in figure F.1.

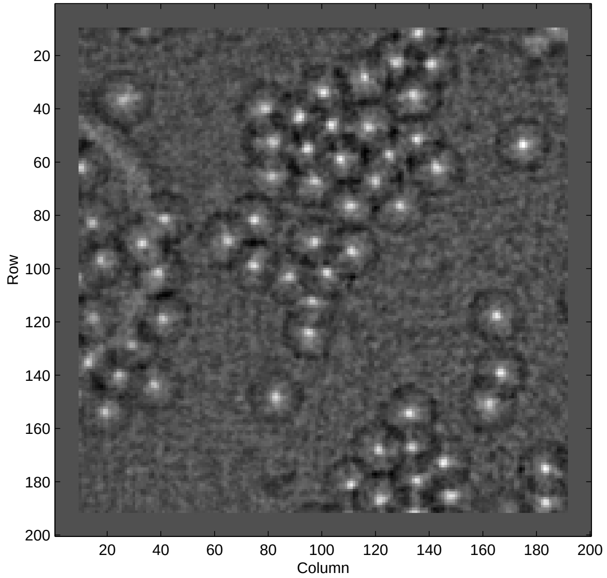


Figure F.3: Likelihood image $L_{(5,5,0)}$ created by filtering the original image y_o by $f_L(5,5,0)$. The pixel value (r_c, c_c) is identical to $P(\boldsymbol{y}_o|r_c, c_c, 5, 5, 0)$.

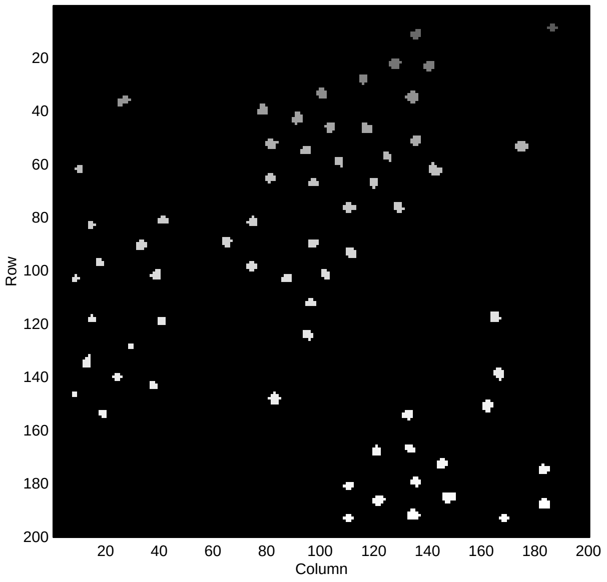


Figure F.4: Likelihood image $L_{(5,5,0)}$ thresholded and labelled.

F.5.2 Optimization

The task of optimization is not only a question of finding the local minimal energy, but also a question of convergence speed. The following performance studies evaluate both aspects. As in the optimization literature the number of function (energy) evaluations measures the speed of convergence. The number of function evaluations is chosen, because the major part of the computation time is spend on function evaluations, so in practice it is a very good approximation of the actual time independent of the specific implementation. All stated energies and number of function evaluations are average values from a large number of particles. For all methods except simulated annealing the stop criterion is defined as $U(\boldsymbol{v}^t, y) - U(\boldsymbol{v}^{t+1}, y) < 0.001$ and the number of function evaluations is limited to 300. As stop criterion for simulated annealing a maximum number of energy evaluation equal to 400 is used, because it is impossible to use $U(\boldsymbol{v}^t, y) - U(\boldsymbol{v}^{t+1}, y) < 0.001$.

The first evaluations are performed to find the optimal method and parameters for the gradient estimation. These tests are performed on noisy initial configurations, i.e. an amount of white noise is added to the initial configurations, generated by the proposed initialization method, to make the optimization more difficult. A steps size $\boldsymbol{h} = (0.5, 0.5, 0.5, 0.5, \pi/20)$ performed well for the forward difference, the central difference and the linear fit gradient estimation methods. In general none of the methods are very sensitive to the step size. The linear fit is made to the energies corresponding to $\boldsymbol{v} + jh_i\boldsymbol{e}_i$ for $j = -2, .., 2$. When the performance of the three methods are compared, the forward difference is 1.8 and 2.9 times faster than the optimization based on the central difference and the linear fit, respectively. But the final average energy is 7-8 percent lower. The final average energy for the central difference and the linear fit differ with less than one percent and the central difference is 1.6 times faster. Due to faster convergence and the small difference in final energy central difference is chosen for gradient estimation in the following test. A comparison of the exact line search and the soft shows that soft line search use less function evaluation and obtain a better final average energy, so the soft line search is used in the following results. The parameters for simulated annealing, which perform best, are $T^0 = 0.01$, $k = 0.98$ and the variance of the random walk $\sigma_{r_c}^2 = \sigma_{c_c}^2 = 0.5, \sigma_a^2 = \sigma_b^2 = 0.25$ and $\sigma_\phi = 0.1$. For Pattern Search an initial step size $h_{ps}^0 = (1, 1, 1, 1, \pi/8)$ and $h_{ps}^0 = (2, 2, 2, 2, \pi/4)$ perform best for the initial and noisy initial configuration, respectively.

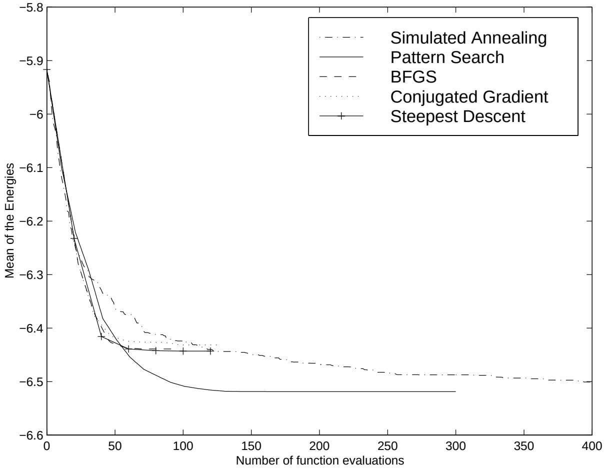


Figure F.5: Performance evaluation for optimization of the initial configuration.

Using the chosen methods and parameters for each optimization algorithm the first evaluation is performed directly on the initial configurations. The general performance results can be seen in table F.1 and figure F.5. Using the generated initial configuration and BFGS for optimization the particles are located, see figure F.6. When the final average energies are compared, all the algorithms perform almost equally well, and a one way analyses of variance rejects the hypotheses that the final energies should be different. In the initial phase of the optimization all the methods converge with the same speed, but during the later optimization Pattern Search and especially Simulated Annealing converge slower, see figure F.5. With respect to the chosen stop criterion the gradient-based methods converge 2.3 faster than Pattern Search, see table F.1. Simulated Annealing is obmitted in this comparison, because it has a different stop criterion. The faster convergence of the gradient based methods is not a surprise, because pattern search and simulated annealing use alot of function evaluations on ”walking in the wrong direction”.

Table F.1: Final average energy, relative energy, average number of function evaluations and relative number of evaluations for optimization of the initial configurations.

Method	SA	PS	SD	CG	BFGS
Energy	-6.49	-6.51	-6.44	-6.43	-6.44
Rel. Energy	1.00	1.00	0.99	0.99	0.99
Std. Energy	0.77	0.79	0.79	0.79	0.78
#Eval.	-	148.5	60.2	62.0	60.5
Rel. #Eval.	-	2.47	1.00	1.03	1.00

The next test is performed on a noisy initial configuration, see table F.2 and figure F.7. With exception of Conjugate Gradient, the final average energies for the methods are approximately the same. This time a one way analysis of variance rejects the hypotheses that the final energies are identical, and a following paired t-test shows that Conjugate Gradient has a significant higher energy than all the other methods. Compared to the previous test, all the final average energies are higher, because a few optimized particles are caught in local minima. In the initial phase of the optimization Pattern Search and Simulated Annealing converge faster than the gradient based method, but again Pattern Search and Simulated Annealing are passed during the later phase of the optimization.

If the maximum number of function evaluations for simulated annealing is increased to 10000, simulated annealing obtain a final average energy of 6.51 and 6.60 for the initial configuration and the noisy initial configuration, respectively. Compared to the energies for the other methods this result is very good, but the computational cost is

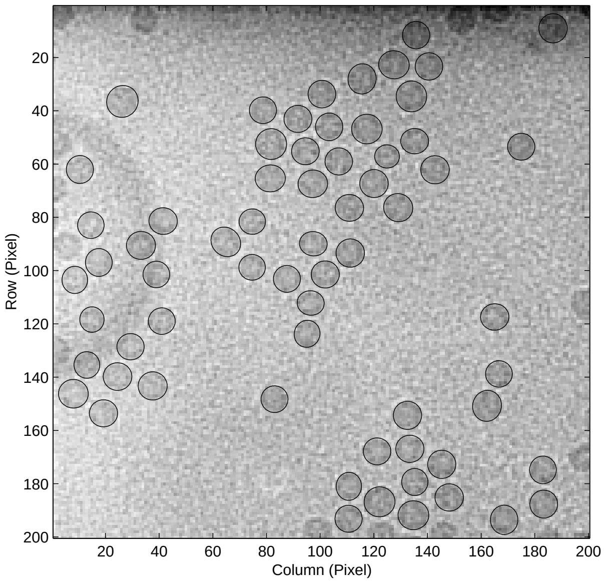


Figure F.6: Estimated particles contours overlaid in original image.

very high. For the noisy configuration Simulated Annealing obtain a statistically significant lower energy than the other methods.

Table F.2: Same measures as table F.1 for optimization of the noisy initial configuration.

Method	SA	PS	SD	CG	BFGS
Energy	-6.33	-6.32	-6.33	-5.83	-6.27
Rel. Energy	1.00	1.00	1.00	0.92	0.99
Std. Energy	0.93	0.97	0.97	1.14	1.05
#Eval.	-	158.0	88.3	92.3	92.0
Rel. #Eval.	-	1.79	1.00	1.05	1.04

F.6 Conclusion and discussion

We have proposed a faster method for initialization of deformable models and introduced a number of algorithms from the optimization literature, which are novel in the context of deformable models. The methods are all general in the sense, that it should be possible to apply them to almost any deformable model.

The proposed method for initialization is significantly faster on the test data than the method based on direct calculations. The method is based on a filter interpretation of the likelihood and presumes that the likelihood can be interpreted as a convolution, which is almost always the case. It should be stressed that the proposed method only is of interest, when part of the initialization task is to find the position of the template. Compared to the initialization method based on direct calculations the only additional work is to created the likelihood filters.

With respect to the final energy the optimization tests show, that all methods perform almost equally well within a reasonable number of function evaluations, but if the number of function evaluations is of no concern simulated annealing obtains very good energies, when it is allowed to use a large number of evaluations. When the speed of convergence is compared, the gradient based methods are fastest during the later phase of the optimization. Compared to the gradient based methods, the relative performance of Pattern Search and Simulated Annealing increase, when the initial configuration is further away from the final solution. In general it is not very satisfactory that some models get caught in local minima and approaches like smoothing the likelihood, stronger regularized models or more prior knowledge would maybe solve part of the problem.

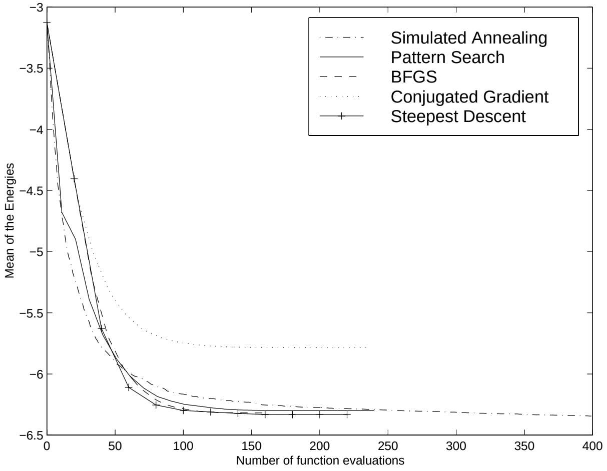


Figure F.7: Performance evaluation for optimization of the noisy initial configuration.

Acknowledgements

We would like to acknowledge Torsten Dörge for the initial work on the nanoparticle model and the Mösbauer group at the Department of Physics, Technical University of Denmark, for the nanoparticle images.

Appendix G

The Grenander Deformable Template Model: A General Scheme

Rune Fisker, Nette Schultz[a], Nicolae Duta[b] and Jens Michael Carstensen[a]
[a] Dep. of Mathematical Modelling, Tech. University of Denmark
[b] Dep. of Computer Science, Michigan State University, USA

Abstract

General deformable models have reduced the need for hand crafting new models for every new problem. But most of the general models still rely on manual interaction by an expert, when applied to a new problem, e.g., for selecting parameters and initialization. In this paper we propose a full and unified scheme for applying the general deformable template model proposed by Grenander et al. [85, 114] to a new problem with minimal manual

interaction, which can be done by a naive user. The main contributions compared to previous work are two learning schemes for the model parameters, a very fast general initialization algorithm and an adaptive likelihood model based on local means. The learning schemes for the model parameters are either based on a training set or just one example. The model parameters are trained by a combination of a 2D shape learning algorithm, a Maximum Likelihood based criteria and a modified version of the Expectation-Maximization algorithm. The fast initialization algorithm is based on a search approach using a filter interpretation of the likelihood model.

Key words: deformable templates, active contour models, initialization, model parameter estimation.

G.1 Introduction

The deformable template model literature is very rich in different models, see [18, 101, 132] for a survey of the best known models. One reasons for this is a general tendency to hand craft a new model for every new problem, even though existing general models, such as [18, 35, 41, 85, 102, 163], can be applied in many cases. Even though general deformable models reduce the need to hand craft a new model, most of the general models still suffer from non-general initialization methods and rely on manual selection of some or all of the model parameters. The model parameters, $\boldsymbol{\theta}$, are defined as the parameters which determine the properties of the optimization criteria. The ultimate goal must be a full automatic algorithm which can be applied to a new problem with no interaction at all. A more realistic goal is to supply one example, and run an incremental or iterative learning algorithm such as the Expectation-Maximization [51]. No matter whether a training set is created manually or by some iterative algorithm, a method for estimation of the model parameters,

$\boldsymbol{\theta}$, based on a training set and a general initialization/optimization method are needed as well. In this paper we proposed such methods for the general deformable template model proposed by Ulf Grenander et al. [85, 114] and further investigated and developed in [73, 86, 96, 109, 151, 155].

G.2 The Grenander model.

The Grenander Model is formulated in the Baysian framework. To make inference the posterior distribution, $P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta})$, is maximized, where $\boldsymbol{v}$ is the template parameters defining the object, $\boldsymbol{y}$ is the image and $\boldsymbol{\theta}$ is the model parameters. Using Bayes theorem, $P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta}) \propto P(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{\theta})P(\boldsymbol{v}|\boldsymbol{\theta})$, the posterior can be separated into the prior, $P(\boldsymbol{v}|\boldsymbol{\theta})$, and the likelihood, $P(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{\theta})$. In the Grenander Model the object is represented by a set of vertices $\boldsymbol{v} = (\boldsymbol{v}_0^T, ..., \boldsymbol{v}_{n-1}^T)^T \in \Omega_v \subseteq \mathbb{R}^{2n}$, where $\boldsymbol{v}_i = (x_i, y_i)^T$. From the vertices $\boldsymbol{v}$ the corresponding edges $\boldsymbol{z} \in \mathbb{R}^{2n}$ can be calculated, where $\boldsymbol{z}_i = \boldsymbol{v}_i - \boldsymbol{v}_{i-1}$ in a cyclic manner (see Figure G.1). The template vector cycle $\boldsymbol{z} = (\boldsymbol{z}_0^T, ..., \boldsymbol{z}_{n-1}^T)^T \in \mathbb{R}^{2n}$ must satisfy the closure constrain: $\sum_{i=0}^{n-1} \boldsymbol{z}_i^{(x)} = 0$ and $\sum_{i=0}^{n-1} \boldsymbol{z}_i^{(y)} = 0$, where $\boldsymbol{z}_i = (z_i^{(x)}, z_i^{(y)})$. In general an object can either be represented by the vertices $\boldsymbol{v}$ or by the edges $\boldsymbol{z}$ and an offset $\boldsymbol{v}_{n-1}$. The relationship is:

$$\boldsymbol{v} = E\boldsymbol{v}_{n-1} + F\boldsymbol{z} \tag{G.1}$$

where

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \vdots & \vdots \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

G.2.1 Prior model.

Assume that a mean or prototype shape $\bar{\boldsymbol{v}} = (\bar{\boldsymbol{v}}_0^T, ..., \bar{\boldsymbol{v}}_{n-1}^T)^T \in \Omega_v$ exists. Given a new shape $\boldsymbol{v}$ with the same number of vertices, the corresponding change in local scale and orientation of each mean edge $\boldsymbol{z}_i = S_i(\kappa_i, \phi_i)\bar{\boldsymbol{z}}_i$ can be determined. Matrices changing scale and orientation are:

$$\begin{bmatrix} \gamma & 0 \\ 0 & \gamma \end{bmatrix}, \gamma \in \mathbb{R}_+ \tag{G.2}$$

and

$$\begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix}, \varphi \in [-\pi, \pi] \tag{G.3}$$

Their product can be written in the following linear form creating the local deformation matrix:

$$S_i(\kappa_i, \phi_i) = \begin{bmatrix} 1+\kappa_i & -\phi_i \\ \phi_i & 1+\kappa_i \end{bmatrix} \tag{G.4}$$

where $\kappa_i = \gamma\cos(\varphi) - 1$ and $\phi_i = -\gamma\sin(\varphi)$. For small values of φ and γ near 1, $\kappa_i \approx \gamma - 1$ controls changes in scale and $\phi_i \approx -\varphi$ controls changes in orientation.

One of the central concepts in the Grenander model is, that the parameters $\boldsymbol{\kappa} = (\kappa_0, ..., \kappa_{n-1})$ and $\boldsymbol{\phi} = (\phi_0, ..., \phi_{n-1})$ are assumed to follow an independent cyclic first-order Gaussian-Markov process with mean zero [85]. The density of a first-order Gaussian-Markov process with mean zero for the n-cyclic parameter vector $\boldsymbol{\kappa}$ is defined by:

$$f(\boldsymbol{\kappa}) = \frac{1}{\sqrt{2\pi}^n \sqrt{|R_\kappa|}} \exp\{-\frac{1}{2}\boldsymbol{\kappa}^T R_\kappa^{-1} \boldsymbol{\kappa}\} \tag{G.5}$$

where R_κ^{-1} is a cyclic tridiagonal band matrix which is positive definite. It follows directly that $\boldsymbol{\kappa} \sim N(\mathbf{0}, R_\kappa)$, where $N(\mathbf{0}, R_\kappa)$ is the n-dimensional multi-variate Gaussian distribution with mean zero $\mathbf{0}$ and covariance R_κ. R_κ^{-1} can be parameterized in the following manner:

$$R_\kappa^{-1} = \begin{bmatrix} \alpha_0 + 2\alpha_1 & -\alpha_1 & 0 & \dots & -\alpha_1 \\ -\alpha_1 & \alpha_0 + 2\alpha_1 & -\alpha_1 & \dots & 0 \\ 0 & -\alpha_1 & \alpha_0 + 2\alpha_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -\alpha_1 & 0 & 0 & \dots & \alpha_0 + 2\alpha_1 \end{bmatrix} \tag{G.6}$$

$\boldsymbol{\alpha} = (\alpha_0,\ \alpha_1)$ are such that R_κ^{-1} is positive definite. This parameterization gives a more intuitive interpretation of the parameters controlling R_κ^{-1} than the original parameterization in [85]. The parameters $\boldsymbol{\alpha}$ are easier to interpret in the following form of the density (G.5):

$$f(\boldsymbol{\kappa}) = \frac{1}{z}\exp\{-\frac{1}{2}\sum_{i=0}^{n-1}(\alpha_0(\kappa_i)^2 + \alpha_1(\kappa_i - \kappa_{i-1})^2)\} \tag{G.7}$$

where $z = 2\pi^{n/2}|R_\kappa|^{1/2}$. α_0 controls the likelihood of how much κ_i differs from zero, see (G.7). For large values of α_0, $f(\boldsymbol{\kappa}) \approx 0$ even for small values of κ_i. In this case it is very unlikely that κ_i differs very much from zero. α_1 controls the neighbor relationship between κ_i and κ_{i+1}. For positive values of α_1 neighbors tend to be positively correlated and the opposite is true for negative α_1. Large values of α_1 correspond to tight bonding between neighbors.

An equivalent density distribution is derived for the orientation parameter vector ϕ with weight parameters α_0 and α_1. In the original formulation individual weights, $\alpha^\kappa_{0,i}, \alpha^\kappa_{1,i}, \alpha^\phi_{0,i}$ and $\alpha^\phi_{1,i}$, are assigned to each κ_i and ϕ_i, but when the model is actually applied all weights are assumed to be equal $\alpha^\kappa_{0,i} = \alpha^\phi_{1,i} = \alpha_0$ and $\alpha^\kappa_{1,i} = \alpha^\phi_{1,i} = \alpha_1$, see [86, 109, 155]. To simplify the notation the index is omitted initially. Recall $\boldsymbol{z}_i = S_i(\kappa_i, \phi_i)\bar{\boldsymbol{z}}_i$. This can be rearranged into:

$$\boldsymbol{z}_i = \begin{bmatrix} \bar{z}^x_i & -\bar{z}^y_i \\ \bar{z}^y_i & \bar{z}^x_i \end{bmatrix} \begin{bmatrix} \kappa_i \\ \phi_i \end{bmatrix} + \bar{\boldsymbol{z}}_i \tag{G.8}$$

Rearranging the global vector cycle this way gives:

$$\boldsymbol{z} = G\boldsymbol{\omega} + \bar{\boldsymbol{z}} \quad \text{for} \quad G = \Gamma H \tag{G.9}$$

where $\boldsymbol{\omega} = (\boldsymbol{\kappa}^T, \boldsymbol{\phi}^T)^T$,

$$\Gamma = \begin{bmatrix} \bar{z}^x_0 & -\bar{z}^y_0 & 0 & 0 & \dots \\ \bar{z}^y_0 & \bar{z}^x_0 & 0 & 0 & \dots \\ 0 & 0 & \bar{z}^x_1 & -\bar{z}^y_1 & \dots \\ 0 & 0 & \bar{z}^y_1 & \bar{z}^x_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{G.10}$$

and H is a permutation matrix interchanging rows:

$$H = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{G.11}$$

If $\boldsymbol{\kappa} \sim N(\mathbf{0}, R_\kappa)$ and $\boldsymbol{\phi} \sim N(\mathbf{0}, R_\phi)$ are stochastically independent, the following distribution is obtained from (G.9) for the edges $\boldsymbol{z} \sim N(\bar{\boldsymbol{z}}, R_z)$ where

$$R_z = GR_\omega G^T \quad \text{for} \quad R_\omega = \begin{bmatrix} R_\kappa & 0 \\ 0 & R_\phi \end{bmatrix} \tag{G.12}$$

Unfortunately this distribution $\boldsymbol{z} \sim N(\bar{\boldsymbol{z}}, R_z)$ does not satisfy the closure constraint, i.e. a sample drawn from $N(\bar{\boldsymbol{z}}, R_z)$ will in general not be closed. The closure constraint can be rewritten to $L\boldsymbol{z} = \mathbf{0}$, where

$$L = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & ... \\ 0 & 1 & 0 & 1 & 0 & 1 & ... \end{bmatrix} \in \mathbb{R}^{2\times n} \tag{G.13}$$

Using the theory on conditioning Gaussian distributions the template vector distribution conditioning on closure is $[\boldsymbol{z}|L\boldsymbol{z} = \mathbf{0}] \sim N(\bar{\boldsymbol{z}}, R)$ where

$$R = R_z - R_z L^T (LR_z L^T)^{-1} LR_z \tag{G.14}$$

The final distribution of the vertices, $\boldsymbol{v}$, is:

$$\boldsymbol{v} \sim N(E\boldsymbol{v}_{n-1} + F\bar{\boldsymbol{z}}, FRF^T) \tag{G.15}$$

Therefore, we consider the prior model, $P(\boldsymbol{v}|\bar{\boldsymbol{v}}, \boldsymbol{\alpha})$, to be distributed $N(E\boldsymbol{v}_{n-1} + F\bar{\boldsymbol{z}}, FRF^T)$.

G.2.2 Likelihood model with global mean.

The original likelihood (or observation) model $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})$ is based on the assumption that the pixel values, $y_{r,c}$, inside and outside the object are independently Gaussian distributed with mean μ_{in} and μ_{out} and common variance λ. This assumption leads to following likelihood:

$$\begin{aligned} P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta}) = & \prod_{(r,c)\in\Omega_{in}} \frac{1}{\sqrt{2\pi\lambda}} \exp\{-\frac{(y_{r,c} - \mu_{in})^2}{2\lambda}\} \times \\ & \prod_{(r,c)\in\Omega_{out}} \frac{1}{\sqrt{2\pi\lambda}} \exp\{-\frac{(y_{r,c} - \mu_{out})^2}{2\lambda}\} \end{aligned} \tag{G.16}$$

where Ω_{in} and Ω_{out} are the set of pixels, which are inside and outside the template, respectively. λ can also be interpreted as a regularization weight parameter, which determines the relative influence of the prior and the likelihood part, respectively. In the original model, the pixels outside are defined as the rest of the image, but for images with varying background it is more reasonable to limit the pixels outside to belong to a local area around the template. In practice, we define the local area by a rectangle a_{out}, which is obtained by a vertical ρ_v and horizontal ρ_h inflation of the smallest rectangle, which contains the actual initial configuration.

G.3 Model parameter estimation based on a training set.

To actually apply the Grenander model values need to be assigned to the model parameters $\boldsymbol{\theta} = (\bar{\boldsymbol{v}}, \boldsymbol{\alpha}, \mu_{in}, \mu_{out}, \lambda, \rho_v, \rho_h)$. In the following sections a fully automated scheme for estimation of the model parameters based on m manually marked object outlines, $\boldsymbol{o}_0, .., \boldsymbol{o}_{m-1}$, in the images, $\boldsymbol{y}_0, ..., \boldsymbol{y}_{m-1}$, is presented. The outlines (manually drawn contours) are not assumed to be registered and are densely (1-pixel) sampled. Compared to a point-based training set with established point correspondences, the outlines are more robust and easier to create, primarily because points do not have to be placed at corresponding positions on each object. Along with the outlines the rectangles, $\boldsymbol{a}_{out,0}, ..., \boldsymbol{a}_{out,m-1}$, defining the local areas need to be given.

The central assumption in the estimation of the model parameters of the prior model is that each training sample corresponds to a sample from the prior model (G.15), which has been changed by a random Euclidean transformation.

G.3.1 Prior mean shape

The crucial step in the estimation of the mean template is to create the aligned parameterization $\boldsymbol{v}_0^t, .., \boldsymbol{v}_{m-1}^t$ of the outlines $\boldsymbol{o}_0, .., \boldsymbol{o}_{m-1}$. Note that $\boldsymbol{v}_0^t, .., \boldsymbol{v}_{m-1}^t$ are aligned with respect to the Euclidean transformation to remove the random Euclidean changes. The mean template $\bar{\boldsymbol{v}}^t$ is then calculated as the simple average of the vertices $\boldsymbol{v}_0^t, .., \boldsymbol{v}_{m-1}^t$.

The parameterization implies a subsampling and a registration process. The simple solution to the problem is to subsample the outline by n equidistant vertices and then perform a point registration (see [129] for an overview of point registration). Unfortunately the separation of subsampling and registration tend to give noisy results.

A 2D Shape Learning algorithm proposed by Duta et al. [59] solves the problem by combining subsampling and registration into one step. One of the main differences from previously reported methods is the manner in which registered points are extracted from each shape outline, using a flexible point matching technique that takes into account both pose/scale differences as well as non-linear shape differences.

The shape learning algorithm produces a set of contours with established point correspondence. The templates in the training set are aligned with the last template and then averaged using Procrustes analysis [56, 83]. This does not insure that the average shape has an average size/orientation with respect to the training set of templates. These properties are obtained by scaling and rotating the output from the shape learning algorithm by $\frac{1}{m}\sum_{i=0}^{m-1}\frac{1}{\gamma_i}$ and $-\frac{1}{m}\sum_{i=0}^{m-1}\varphi_i$, where γ_i and φ_i are the scale and rotation, which aligned shape i with the last shape.

G.3.2 Prior weight parameters

Most deformable models contain weight parameters, but never the less weight parameter estimation has received very little attention in the literature. With few exceptions the weight parameters are tuned manually. The work by Lai et al. [118] is one exception. They use a non-general minimax criteria to estimate the regularization parameter. Fisker & Carstensen [65] uses a combined maximum likelihood and minimum distance criteria to estimate weight parameters in a deformable model used for textile inspection. Recent work by Kent et al. [109] and Hurn et al. [96] has considered the question of weight parameter estimation in Grenander related models. The work by Kent et al. [109] only considers circulant symmetry objects and the work by Hurn et al. [96] uses a closure constraint based on $\boldsymbol{v}_0 = \boldsymbol{v}_n$. Unfortunately this closure constraint leads to an inhomogeneous covariance structure, where the vertices in the end of the vector cycle have much higher variance, than those in the start.

This should be compared to the covariance in (G.15), which has an almost homogeneous covariance structure.

The estimation of the weight parameters are based on the aligned training samples, $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{m-1}^t$. Using the assumption that $\boldsymbol{v}_i^t$ is a sample from the prior distribution $N(E\boldsymbol{v}_{n-1} + F\bar{\boldsymbol{z}}, FRF^T)$, we propose to estimate the weights $\boldsymbol{\alpha}$ using the Maximum Likelihood (ML) estimator:

$$\hat{\boldsymbol{\alpha}} = \max_{\boldsymbol{\alpha}} \prod_{i=0}^{m-1} P(\boldsymbol{v}_i^t|\bar{\boldsymbol{v}}^t, \boldsymbol{\alpha}) \tag{G.17}$$

To evaluate $P(\boldsymbol{v}_i|\bar{\boldsymbol{v}}, \boldsymbol{\alpha})$ the inverse of FRF^T has to be calculated. Unfortunately this is not straight forward, because R is rank deficient, due the closure constraint, $\sum_{i=0}^{n-1}\boldsymbol{z}_i = \boldsymbol{0}$, which removes two degrees of freedom, i.e. $\text{rank}(FRF^T) = 2n-2$. This problem is solved by using a pseudo inverse of FRF^T. Due to the closure constrain the two last rows and columns of FRF^T will always be zero. This is basically also the reason for the last element of $\boldsymbol{v}$ being identical to the offset $\boldsymbol{v}_{n-1}$ in (G.1). Based on this observation the pseudo inverse $(FRF^T)^-$ is defined by:

$$(FRF^T)^- = \begin{bmatrix} \Sigma_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{G.18}$$

where Σ_{11} is the first $2n-2$ rows and columns of FRF^T. The pseudo inverse is also used, when the prior is evaluated.

G.3.3 Likelihood parameters

The means, μ_{in}, μ_{out}, and the common variance, λ, used in the likelihood model can be estimated from $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{m-1}^t$, $\boldsymbol{a}_{out,0}, ..., \boldsymbol{a}_{out,m-1}$ and the corresponding images, $\boldsymbol{y}_0, ..., \boldsymbol{y}_{m-1}$, by calculating the traditional means and the pooled variance of the pixels values inside and outside the template. In fact this is also a Maximum Likelihood estimate.

The true vertical ρ_v and horizontal ρ_h inflation can not be estimated directly, so ρ_v and ρ_h are approximated by the average vertical and horizontal inflation, which map the smallest rectangle containing $\boldsymbol{v}_i^t$ to $\boldsymbol{a}_{out,i}$ independent of position. This approximation is based on the assumption, that the initial configuration will be placed close to the center of $\boldsymbol{a}_{out,i}$ with approximately the same size as $\boldsymbol{v}_i^t$.

G.4 Initialization.

To actually make inference about an object in an image y, estimation of the maximum a posteriori (MAP), $\hat{\boldsymbol{v}} = \max_{\boldsymbol{v}} P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta})$, is performed. This MAP estimation is usually separated into two steps: Initialization and Optimization.

Most deformable models have been initialized using heuristics tuned to the specific problem, which is not acceptable from the general point of view. Different initialization approaches using the mode of the prior [163], the generalized Hough transform [72, 73, 118] and moments [18] have been applied with success to the respective models, but these approaches cannot be applied in the general setting of the Grenander Model. The only general initialization approach, which we have knowledge about, is the search strategy [61, 102], where a sparse search is performed in the parameter space Ω_v. Note, that the Hough transform [95, 98, 122] in general can be interpreted as a deformable model with a uniform prior, where the initialization is performed by an intensive and efficient search and the optimization is omitted.

For most problems the initial position is unknown, and a computational expensive part of the deterministic search is to shift different configurations of the template around the image and calculate the posterior probability $P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta})$. Based on this observation we proposed a fast search strategy using a filter interpretation, $f_l(\boldsymbol{v},\boldsymbol{\theta})$, of the likelihood energy $U(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta})$, where $P(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{\theta}) = \frac{1}{c}\exp\{-U(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta})\}$, such that the likelihood energy, which corresponds to the template positioned with the center at each pixel of the image, can be calculated by a correlation of the filter $f_l(\boldsymbol{v},\boldsymbol{\theta})$ and the image y. The likelihood energy $U(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta})$, derived directly from $P(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{\theta}) = \frac{1}{c}\exp\{-U(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta})\}$ in (G.16), can be rearranging into a filtering in the following way (note this is still only for one position of the template):

$$\begin{aligned} U(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta}) &= \sum_{\Omega_{in}} \frac{(y-\mu_{in})^2}{2\lambda} + \sum_{\Omega_{out}} \frac{(y-\mu_{out})^2}{2\lambda} \\ &= \sum_{\Omega_{in}\cup\Omega_{out}} \frac{y^2}{2\lambda} + \frac{|\Omega_{in}|}{2\lambda}\mu_{in}^2 + \frac{|\Omega_{out}|}{2\lambda}\mu_{out}^2 - \\ &\quad \sum_{\Omega_{in}} \frac{y\mu_{in}}{\lambda} - \sum_{\Omega_{out}} \frac{y\mu_{out}}{\lambda} \\ &= \sum_{\Omega_{in}\cup\Omega_{out}} \frac{y^2}{2\lambda} + \frac{|\Omega_{in}|}{2\lambda}\mu_{in}^2 + \frac{|\Omega_{out}|}{2\lambda}\mu_{out}^2 - \\ &\quad y * f_l(\boldsymbol{v},\mu_{in},\mu_{out},\lambda) \end{aligned} \tag{G.19}$$

where $|\Omega|$ is the number of pixels in Ω, $\sum_{\Omega} y$ is a short form for $\sum_{(r,c)\in\Omega} y(r,c)$, $y * f_l(\boldsymbol{v},\mu_{in},\mu_{out},\lambda)$ is the convolution for one position and $f_l(\boldsymbol{v},\mu_{in},\mu_{out},\lambda)$ is the likelihood interpreted filter, which has the size of a_{out} and contain two constant areas with the coefficients μ_{in}/λ and μ_{out}/λ depending on whether the pixels will be inside or outside the object. To calculate the likelihood energy for the centers of the template corresponding to a region of interest ROI in the image, the convolution of $\boldsymbol{y}$ and $f_l(\boldsymbol{v},\mu_{in},\mu_{out},\lambda)$ is performed for the ROI and the terms $\sum_{\Omega_{in}\cup\Omega_{out}} \frac{y^2}{2\lambda}$ and $\frac{|\Omega_{in}|}{2\lambda}\mu_{in}^2 + \frac{|\Omega_{out}|}{2\lambda}\mu_{out}^2$ are evaluated.

Compared to calculating the likelihood independently at different positions, this approach gives a significant reduction in the practical computational cost. This reduction is mainly due to highly optimized convolution procedures and avoidance of unnecessary recalculations at each position. Another significant gain in computation time can be obtained, if the image can be resampled/paded into having 2^r columns and 2^s rows, respectively, where $r \in \mathbb{N}_+$ and $s \in \mathbb{N}_+$. Then the convolutions can be performed in the Fourier space using the convolution theorem and Fast Fourier Transform. The full initialization algorithm can be summarized as follows:

1. Create relevant search configurations $\boldsymbol{v}_0, ..., \boldsymbol{v}_{k-1}$ and set $i = 0$.
2. Create likelihood interpreted filters $f_l(\boldsymbol{v}_i, \boldsymbol{\theta})$ (and optionally their Fourier transform $F_l(\boldsymbol{v}_i, \boldsymbol{\theta})$).
3. Calculate $P(\boldsymbol{y}|\boldsymbol{v}_i, \boldsymbol{\theta})$ for the center of the template corresponding to each pixel within the ROI by a correlation of $f_l(\boldsymbol{v}_i, \boldsymbol{\theta})$ and $\boldsymbol{y}$.
4. Calculate $P(\boldsymbol{v}_i|\boldsymbol{y}, \boldsymbol{\theta}) = P(\boldsymbol{v}_i|\boldsymbol{\theta})P(\boldsymbol{y}|\boldsymbol{v}_i, \boldsymbol{\theta})$
5. $i = i + 1$. Go to 2 if $i < k$.
6. Compute the initial configurations from $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})$.

The actual choice and number of initial search configurations $\boldsymbol{v}_0, ...,$ $\boldsymbol{v}_{k-1}$ is determined by the amount of variation in scale, orientation and shape of the training set combined with the overall demand for precise initialization. For most problems it is enough to do the search in the Euclidean space, i.e. scale, translation and orientation, but for harder problems configurations with shape variation need to be included. Ways of including shape variation is to apply configurations sampled from the prior model or configurations created from the mean shape and the most important eigenvectors of the empirical or imposed covariance structure. In the normal setting the ROI is chosen, such that the full template always is inside the image.

The final step is to extract the initial configurations from the calculated $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})$. In practice this is done by extracting the maximum posterior $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})_{(r,c)}$ corresponding to each template center (r, c) based on the assumption that only one template has the center in (r, c). In the case where the number of objects in the image is known, the initial configurations are extracted as the corresponding number of local maxima with the highest posterior probability $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})_{(r,c)}$. In the case where the number of objects in the image is unknown the initial configurations are extracted as local maximas with a posterior probability $P(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\theta})_{(r,c)}$ above a threshold t_p.

G.5 Optimization

The optimization is performed using a Simulated Annealing scheme [27, 112] incorporating the Metropolis algorithm [134]. The temperature is decreased by an exponential temperature scheme $T_{t+1} = k_T T_t$ and the new sample $\boldsymbol{v}_{new}$ is generated by sampling in the prior distribution $P(\boldsymbol{v}|\boldsymbol{\theta})$. In practice only a small segment $\boldsymbol{v}_{seg} = (\boldsymbol{v}_l^T, ..., \boldsymbol{v}_{l+q-1}^T)^T$ of the vector cycle is changed at each iteration. The samples are then generated from the conditional prior distribution $P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{/seg}, \boldsymbol{\theta})$, where $\boldsymbol{v}_{/seg} = (\boldsymbol{v}_0, ..., \boldsymbol{v}_{l-1}, \boldsymbol{v}_{l+q}, ..., \boldsymbol{v}_{n-1})$. Due to the first-order Markov assumption the conditional distribution only depends on the endpoints of the segment $P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{/seg}, \boldsymbol{\theta}) = P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{l-1}, \boldsymbol{v}_{l+q}, \boldsymbol{\theta})$. In practice the conditional distribution $P(\boldsymbol{v}_{seg}|$ $\boldsymbol{v}_{l-1}, \boldsymbol{v}_{l+q}, \boldsymbol{\theta})$ is obtained from the conditional distributions $f(\boldsymbol{\kappa}_{seg}$ $|\boldsymbol{\kappa}_{l-1}, \boldsymbol{\kappa}_{l+q})$ and $f(\boldsymbol{\phi}_{seg}|\boldsymbol{\phi}_{l-1}, \boldsymbol{\phi}_{l+q})$, which are propagated through a similar framework as used for the full prior distribution in section G.2.1. We refer to [155] for the entire derivation, but note that the means for $f(\boldsymbol{\kappa}_{seg}|\boldsymbol{\kappa}_{l-1}, \boldsymbol{\kappa}_{l+q})$ and $f(\boldsymbol{\phi}_{seg}|\boldsymbol{\phi}_{l-1}, \boldsymbol{\phi}_{l+q})$ applied in this work are the true conditional means created by the theory for conditioning Gaussian distributions and not the approximations used in [155].

Due to the Metropolis algorithm it is only necessary to calculate the ratio between the posteriors $P(\boldsymbol{v}_{new}|\boldsymbol{y}, \boldsymbol{\theta})/P(\boldsymbol{v}_{old}|\boldsymbol{y}, \boldsymbol{\theta})$. Because most of $\boldsymbol{v}_{new}$ and $\boldsymbol{v}_{old}$ are identical the likelihood ratio reduces to:

$$\frac{P(\boldsymbol{y}|\boldsymbol{v}_{new},\boldsymbol{\theta})}{P(\boldsymbol{y}|\boldsymbol{v}_{old},\boldsymbol{\theta})} = \exp\{\sum_{\Omega_{in}^{old}} \frac{(y-\mu_{in})^2}{2\lambda} - \sum_{\Omega_{in}^{new}} \frac{(y-\mu_{in})^2}{2\lambda} + \sum_{\Omega_{out}^{old}} \frac{(y-\mu_{out})^2}{2\lambda} - \sum_{\Omega_{out}^{new}} \frac{(y-\mu_{out})^2}{2\lambda}\}$$

where the sum is performed over the set of pixels, which are influenced by the change in the segment $\boldsymbol{v}_{seg}$. Ω_{in}^{new}, Ω_{out}^{new}, Ω_{in}^{old} and Ω_{out}^{old} separate this set into the pixels which belong to the inside and outside of $\boldsymbol{v}_{new}$ and $\boldsymbol{v}_{old}$, respectively.

G.6 Model parameter estimation based on one example.

Even when outlines are used for the training sets, the manual tracing is a cumbersome task. To minimize the need for manual interaction the following model parameter estimation, based on a modified version of the Expectation-Maximization (EM) algorithm [51], is proposed:

1. Set $j = 0$ and initialize $\boldsymbol{\theta}_j$.
2. Estimate $\hat{\boldsymbol{v}}_{\boldsymbol{i}} = \max_{\boldsymbol{v}_i} P(\boldsymbol{v}_i|\boldsymbol{y}_i, \boldsymbol{\theta}_j)$ for $i = 0, .., m-1$
3. Estimate $\boldsymbol{\theta}_j$ by the training set based model parameter estimation using $\hat{\boldsymbol{v}}_0, ..., \hat{\boldsymbol{v}}_{m-1}$ as the training set.
4. j = j + 1. Go to 2 until stop.

The manual input is now limited to the initialization of the model parameters, $\boldsymbol{\theta}_0$. Note a training set of images is still required. The most qualified way to select the initial values is to manually outline one representative shape, $\boldsymbol{o}_i$, and the local area, $\boldsymbol{a}_{out,i}$, in the corresponding image, $\boldsymbol{y}_i$. A resampled version of the outline, $\boldsymbol{o}_i$, is then used as the initial mean shape, $\bar{\boldsymbol{v}}$, and the initial values of $\mu_{in}, \mu_{out}, \lambda, \rho_v$ and ρ_h are estimated from $\boldsymbol{y}_i$ and $\boldsymbol{a}_{out,i}$. When only one shape outline is given, it is not feasible to estimate an initial value of $\boldsymbol{\alpha}$, so a guess need to be made.

The EM algorithm is shown to converge to the ML estimate at least locally under some moderate conditions [185]. Global convergence is also proven for special cases, e.g. in [120, 149]. However, the convergence of the EM algorithm can be painfully slow [125]. With respect to the proposed algorithm is does not seem to be possible to do a formal proof on the convergence, but the experimental results indicate good convergence.

G.7 Adaptive local mean model.

For many problems the local mean varies over the object and background, and the assumption of one global mean, μ_{in} and μ_{out}, is insufficient. Another problem is that the constant means make the likelihood model very sensitive to changes in the gray level intensity. On this basis we propose a likelihood model, which adapts to the local mean in a band around the edges. The likelihood ratio for the local mean model is:

$$\frac{P(\boldsymbol{y}|\boldsymbol{v}_{new},\theta)}{P(\boldsymbol{y}|\boldsymbol{v}_{old},\theta)} = \exp\{\sum_{\Omega_{in}^{old}} \frac{(y-\mu_{in}^{seg})^2}{2\lambda} - \sum_{\Omega_{in}^{new}} \frac{(y-\mu_{in}^{seg})^2}{2\lambda} + \sum_{\Omega_{out}^{old}} \frac{(y-\mu_{out}^{seg})^2}{2\lambda} - \sum_{\Omega_{out}^{new}} \frac{(y-\mu_{out}^{seg})^2}{2\lambda}\}$$

where μ_{in}^{seg} and μ_{out}^{seg} are the local means inside and outside of the segment, $\boldsymbol{v}_{seg}$. To adapt to the pixel intensities the local means are estimated from the actual configuration $\boldsymbol{v}_{old}$. To apply equal weight to each edge, $\boldsymbol{z}_i$, the means are estimated as the average of the means, $\mu_{i,in}$ and $\mu_{i,out}$, around each edge $\boldsymbol{z}_i$. In practice $\mu_{i,in}$ and $\mu_{i,out}$ are calculated from the pixels in a local band with width b around $\boldsymbol{z}_i$. Only pixels inside a_{out} are taken into account. The size of the band width b is estimated as the average of the bands manually marked around each training sample $\boldsymbol{v}_i^t$. Initially all means inside and outside are assumed to be identical, i.e. $\mu_{i,in} = \mu_{in}$ and $\mu_{i,out} = \mu_{out}$. Based on this assumption the original model is used for initialization following section G.4.

G.8 Experimental results.

The proposed framework has been applied to segmentation of cross-sections of pork carcasses. This is a part of a study of different properties of meat done for a number of Slaughter-houses (see Figure G.2).

To estimate the model parameters, 14 outlines of cross-sections, $\boldsymbol{o}_0, ..., \boldsymbol{o}_{13}$, have been manually marked in 14 images. The first step is to apply the 2D shape learning algorithm [59] to subsample and align the shapes creating the corresponding template parameters $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{13}^t$ (see Figure G.3). The number of vertices has manually been specified to 83, which basically is a reasonable tradeoff between speed and accuracy, see discussion in section G.9. The mean shape, $\bar{\boldsymbol{v}}^t$, is then created as the average of $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{13}^t$ (see Figure G.3). Before the ML estimator (G.17) is used to estimate the empirical weight parameters, the approach has been tested on simulated data with the mean equal to $\bar{\boldsymbol{v}}^t$. In general the results indicate a robust and stable method which is not to sensitive to small sample sizes. The Maximum Likelihood criteria seems to have a smooth convex surface, making the actual optimization very robust.

The empirical weight parameters $\boldsymbol{\alpha}$ are estimated to $\alpha_0 = 22.96$ and $\alpha_1 = 3.57$. A powerful tool to verify the prior model and the

estimated mean, $\bar{\boldsymbol{v}}$, and weights, $\boldsymbol{\alpha}$, is to generate samples from the prior model (G.15) (see Figure G.4). When the simulations are compared to the real shapes in Figure G.3, the simulated shapes show a shape variation, which is reasonably close to variation in the real samples. This simulation basically verifies that the derived prior model with the estimated parameters is an acceptable model of the real shape variation. The last step in the estimation of the model parameters, $\boldsymbol{\theta}$, is the calculation of $\mu_{in} = 150.60, \mu_{out} = 30.69, \lambda = 778.05, \rho_v = 0.18$ and $\rho_h = 0.09$ following section G.3.3.

Based on the limited variation in scale, rotation and shape, the mean shape $\bar{\boldsymbol{v}}^t$ is chosen to be the only relevant configuration used in the fast search strategy. Using the estimated model parameters the likelihood interpreted filter $f_l(\boldsymbol{\theta})$ and its the Fourier transform $F_l(\boldsymbol{\theta})$ is created (see Figure G.5). This completes the off-line training.

Given a new image $\boldsymbol{y}_0$ the first step to make inference is to perform the initialization using the fast search strategy. The core of the fast search strategy is to calculate the likelihood energy for the center of the mean template corresponding to each pixel within the ROI by a correlation of $f_l(\boldsymbol{\theta})$ and $\boldsymbol{y}_0$ (see Figures G.2, G.5 and G.6). The correlation is performed in the Fourier space by temporarily padding the image to 512x1024. The final posterior energy is obtained by adding the prior energy, which in this case is zero. The initial configuration is selected as the configuration, which corresponds to the maximum posterior energy, using the knowledge that there will be one and only one cross-section in the image. Comparing the computation time for the likelihood energy by independent calculations (977.04 sec. on a Pentium II 350 Mhz) and by the correlation approach (1.06 sec.), the latter is approximately 900 times faster, leading to an overall initialization times that is approximately 600 times faster (independent: 977.54 sec., correlation: 1.56 sec.). Note that the independent calculations only are done for the ROI, i.e. 35490 pixel $\sim$ 9 percent of the image. Whereas the Fourier based approach calculate the likelihood energy for the full image using a cyclic border approach. The computation times is of course highly sensitive to the actual implementation, but both implementations are optimized with respect to speed.

The second step is to perform the actual optimization based on the initial configuration. In the simulated annealing scheme is used $T_0 = 50$, $k_T = 0.97$, 400 iterations and a segment size of 3. The methods are evaluated on a test set containing 15 images with manually marked outlines. The average point to associated boundary error [43] measured from the outline to the template is 4.07(±6.30) and 1.02(±0.46) pixels using the global mean and local mean criteria, respectively. The result of optimizing the initial configuration (73.87 sec.) in Figure G.2 using the adaptive local mean criteria is shown in Figure G.7. The very high error for the global mean mainly originate from errors in 3 images, where the pixel intensities differ allot from the estimated average intensities, μ_{in} and μ_{out}. The performance of the global mean is improved to an error of 1.54(±0.48), when an adaptive estimation of μ_{in}, μ_{out} and λ is introduced after the initialization.

The last experiments concern the model parameter estimation based on one example. The initial values of $\boldsymbol{\theta}$, except $\boldsymbol{\alpha}$, are estimated from the representative outline, $\boldsymbol{o}_0$, and the corresponding image $\boldsymbol{y}_0$ and local area $\boldsymbol{a}_{out,0}$. Finally a guess is made on the initial values of $\boldsymbol{\alpha}$. The estimated model parameters are $\boldsymbol{\alpha} = (14.30, 22.60)$, $\mu_{in} = 144.16, \mu_{out} = 38.20, \lambda = 1289.12, \rho_v = 0.15$ and $\rho_h = 0.10$. Compared to the training set based estimate of $\boldsymbol{\alpha}$, the values α_0 has decreased and α_1 increased. However this is no surprise, since the mean shape is both used to segment the cross-sections and iteratively estimated from the segmented cross-sections, which should increase the correlation between edges, i.e. α_1. Due to the diagonal elements, $\alpha_0 + 2\alpha_1$, of the covariance structure in (G.6), it seems correct that α_0 decreases when α_1 increases, if the overall covariance structure should be in reasonably good accordance with the previously estimated. The pixel statistics estimates vary a bit compared to the training set based estimates, which is due to the fact that the local areas are set by the estimated ρ_v and ρ_h, so sometimes parts of the

box below the cross-sections are included in the local area. With respect to convergence, the algorithm usually converges within 40 iterations and does not seem to be very sensitive to the initial guess on $\boldsymbol{\alpha}$. When the estimated parameters are applied to the test set, the error is 1.21(±0.78).

G.9 Conclusion and Discussion.

We have proposed a general scheme for applying the Grenander Model to a new problem, which requires minimal manual interaction. The scheme has successfully been applied to segmentation of cross-sections of pork carcasses. The test results indicate a very fast general search based initialization algorithm, which is significantly faster than a search using independent calculations. We have also proposed an adaptive likelihood model based on local means, which obtain better performance on the test set than the original model. When the performance of the training set based model parameter estimation is compared to the estimation based on one example, the first obtain the best performance, but the estimation based on one example still obtained promising results.

The proposed scheme has reduced the needed for manual selection of parameters significantly, but still the number of vertices, the optimization parameters and the initial values of $\boldsymbol{\alpha}$ need to be selected. The selection of the number of vertices is a tradeoff between speed and accuracy, but in a statistical setting the number of vertices can be selected straight forward by using a model selection criteria like the Akaike Information Criteria [1]. Note that the outline based training sets make it very easy to change the number of vertices, because of the 2D shape learning algorithm. The manual selection of the optimization parameters and the initial values of $\boldsymbol{\alpha}$ is a minor problem, since these parameters are quite general.

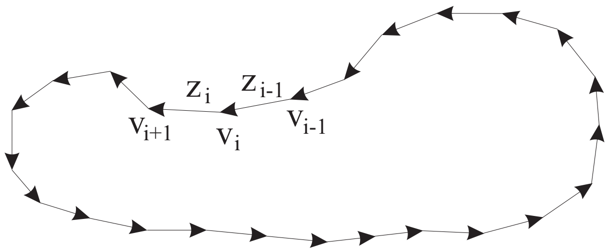


Figure G.1: A template with vertices $\boldsymbol{v}$ and edges $\boldsymbol{z}$.

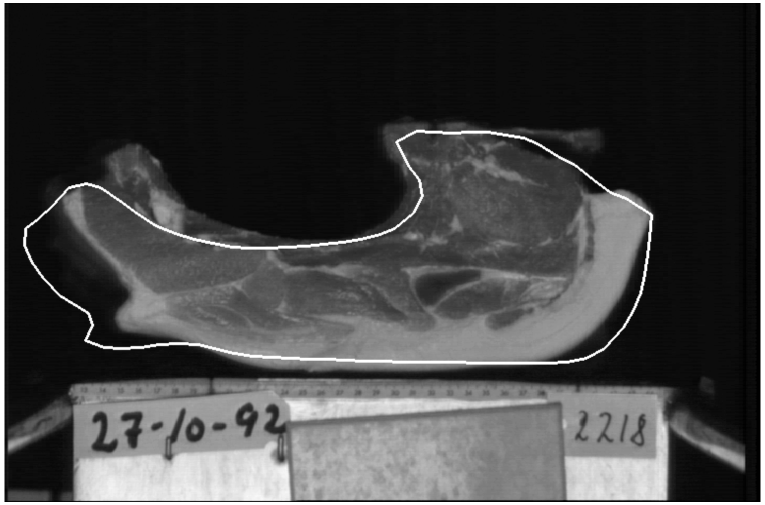


Figure G.2: Image $\boldsymbol{y}_0$ of cross-section of pork carcass (512x768) with initial configuration overlaid.

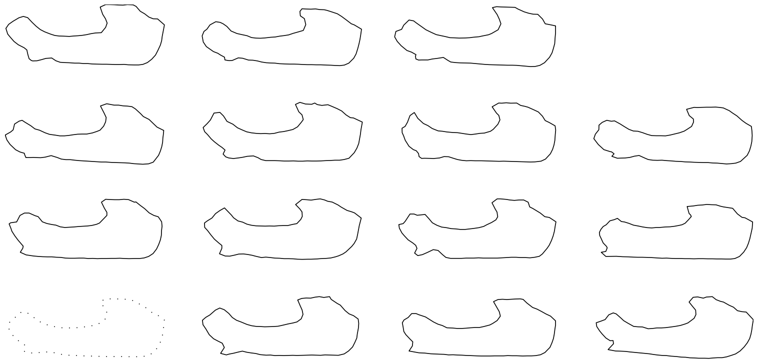


Figure G.3: Subsampled and aligned training set (full line) and mean shape (dotted line).

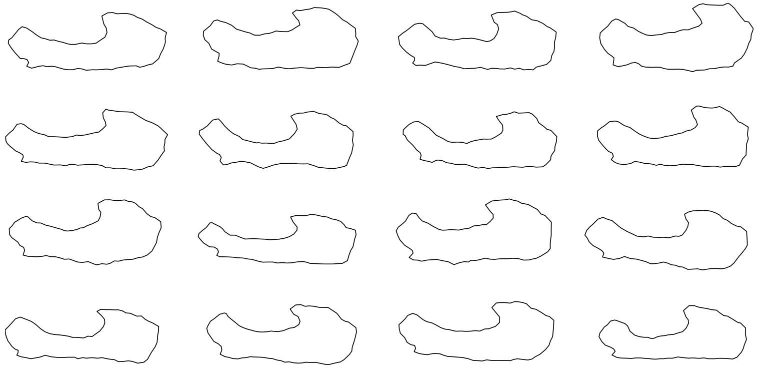


Figure G.4: Shape simulations using estimated $\boldsymbol{\alpha}$ and $\bar{\boldsymbol{v}}$.

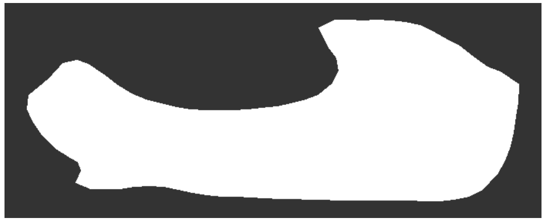


Figure G.5: Likelihood interpreted filter $f_l(\boldsymbol{\theta})$ (283x696). The bright and dark color correspond to the pixels inside and outside $\bar{\boldsymbol{v}}^t$ limited by the local area a_{out}.

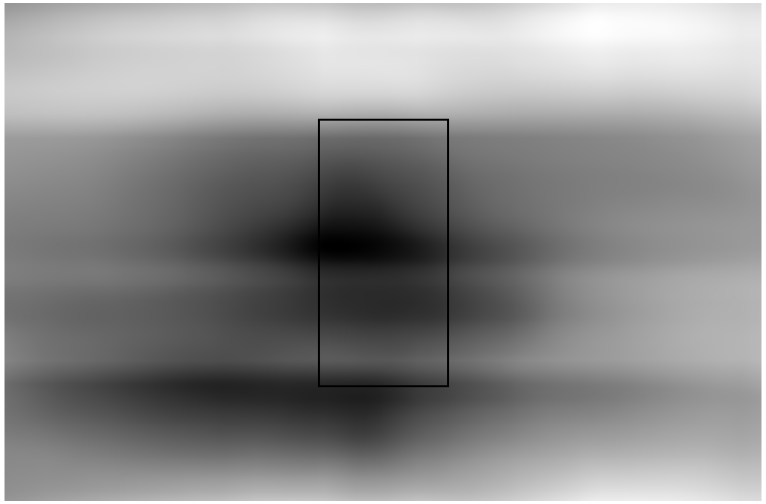


Figure G.6: Likelihood energy image. The pixel (r, c) is identical to $U_l(\boldsymbol{y}_0|\bar{\boldsymbol{v}}^t, \boldsymbol{\theta})$ where $\bar{\boldsymbol{v}}^t$ has center in (r, c). The black rectangle corresponds to the ROI.

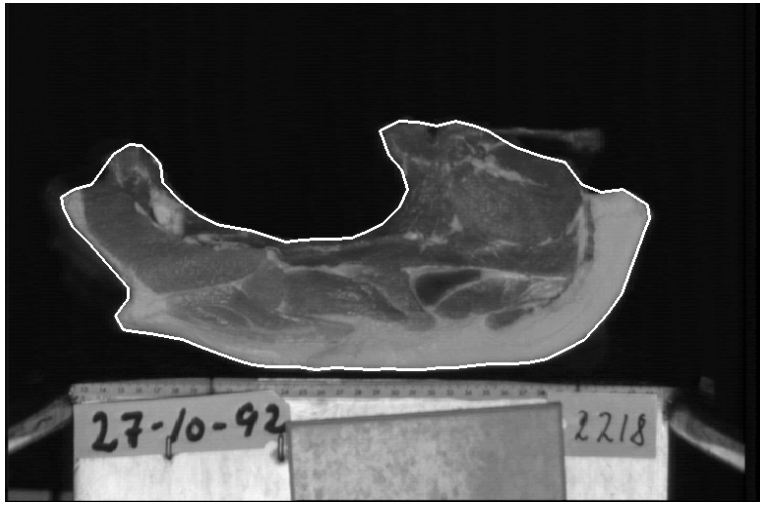


Figure G.7: Optimized template using local means.

Bibliography

[1] H. Akaike. A new look at statistical model identification. *IEEE trans. Automatic Control*, 19:716–723, 1974.

[2] J. Altmann and H. J. P. Reitbock. A fast correlation method for scale- and translation invariant pattern recognition. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 1(6):46–57, 1984.

[3] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *Jour. of Optimization Theory and Applications*, 47(1):1–16, 1985.

[4] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 12(9):855–867, 1990.

[5] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration thrugh deformable templates. *Jour. of the American Statistical Assocoation.*, 86(414):376–387, 1991.

[6] P. R. Andresen and M. Nielsen. Non-rigid registration by geometry-constrained diffusion. *Lecture Notes in Computer Science*, 1679:533–543, 1999.

[7] N. Ayache. Medical computer vision, virtual reality and robotics. *Image and Vision computing*, 13(4):295–313, 1995.

[8] J.-C. Bacri, R. Perzynski, D. Salin, V. Cabuil, and R. Massart. Ionic ferrofluids: A crossing of chemistry and physics. *J. Magn. Magn. Mater.*, 85:27–32, 1990.

[9] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 61(1):1–21, 1989.

[10] R. Bajcsy, R. Lieberson, and M. Reivich. A computerized system for the elastic matching of deformed radiographic to idealized atlas images. *J. Comput. Assisted Tomog.*, 7(4):612–625, 1983.

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

[12] L. Ballerini. Genetic snakes for medical image segmentation. In *Proc. of the SPIE*, volume 3457, pages 284–295, 1998.

[13] D. P. Bertsekas. *Dynamic Programming.* Prentice-Hall, 1987.

[14] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

[15] J. Besag. Towards bayesian image analysis. *Jour. of Applied Statistics*, 16(3):395–407, 1989.

[16] A Del Bimbo and P Pala. Visual image retrieval by elastic matching of user sketches. *Computer Vision and Image Understanding*, 19(2):121–132, 1997.

[17] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. Journal of Computer Vision*, 19(1):57–92, 1996.

[18] A. Blake and M. Isard. *Active Contours.* Springer, 1998.

[19] F. L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–85, 1989.

[20] M. Bro-Nielsen and C. Gramkow. Fast fluid registration of medical images. In *Proc. Visualization in Biomedical Computing*, pages 267–276, 1996.

[21] L. G. Brown. Survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[22] D. J. Burr. Elastic matching of line drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3(6):708–713, 1981.

[23] J. M. Carstensen. An active lattice model in a Bayesian framework. *Computer Vision and Image Understanding*, 63(2):380–387, 1996.

[24] J. M. Carstensen, M. Grunkin, and K. Conradsen. Measurement of enzymatic treatment effect on textile using digital image analyse. In *Proc. MVA '92 IAPR Workshop on Machine Vision Applications*, pages 277–280, Tokyo; Japan, 1992.

[25] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–33, 1993.

[26] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Jour. of Computer Vision*, 22(1):61–79, 1997.

[27] V. Cerny. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Jour. of Optimization Theory and Applications*, 45:41–51, 1985.

[28] S. Chandran and A.K. Potty. Energy minimization of contours using boundary conditions. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 20(5):546–549, 1998.

[29] G. E. Christensen. *Deformable Shape Models for Anatomy.* PhD thesis, Sever Institute of Tech., Washington University, Saint Louis, Mossouri, 1994.

[30] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE Trans. on Image Processing*, 4(10):1435–1447, 1996.

[31] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. 3th Int. Conf. on Computer Vision*, pages 616–623, 1990.

[32] Fernand S. Cohen, Zhigang Fan, and Stephane Attali. Automated inspection of textile fabrics using textural models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(8):803–808, August 1991.

[33] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[34] 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 Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[35] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. European Conf. on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[36] T. F. Cootes and C. J. Taylor. Active shape models-'smart snakes'. In *Proc. British Machine Vision Conf., BMVC92*, pages 266–275, 1992.

[37] T. F. Cootes and C. J. Taylor. Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing*, 13(5):403–9, 1995.

[38] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–574, 1999.

[39] T. F. Cootes and C. J. Taylor. Combining elastic and statistical models of appearance variation. In *ECCV 2000, Proc. European Conference on Computer Vision*, volume 1, pages 149–163, 2000.

[40] T. F. Cootes and C. J Taylor. *Statistical Models of Appearance for Computer Vision.* Tech. Report , University of Manchester, http://s10d.smb.man.ac.uk/ bim/index.html, Feb. 2000.

[41] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[42] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active shape models: Evaluation of a multi-resolution method for improved image search. In *Proc. British Machine Vision Conf.*, pages 327–336, 1994.

[43] T.F. Cootes, G. J. Edwards, and C. J. Taylor. Comparing active shape models with active appearance models. In *Proc. British Machine Vision Conf.*, pages 173–182, 1999.

[44] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* The MIT Press, 1990.

[45] J. Coughlan, A. Yuille, C. English, and D. Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000.

[46] P. Craven and G. Wahba. Smoothing noisy data with spline functions - estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[47] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, 41:909–996, 1988.

[48] I. Daubechies. *Ten Lectures on Wavelets.* Soc. for Industrial and Applied Math., 1992.

[49] C. Davatzikos. Spatial transformation and registration of brain images using elastically deformable models. *Computer Vision and Image Understanding*, 66(2):207–222, 1997.

[50] K. Delibasis and P.E. Undrill. Genetic algorithms and deformable geometric models for anatomical object recognition. In *IEE Coll. on 'Genetic Algorithms in Image Processing and Vision'*, pages 8/1–8/7, 1994.

[51] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Jour. Royal Statistical Society, Series B*, 39:1–38, 1977.

[52] J. E. Dennis and R. B. Schnabel. *Numerical Methods For Unconstrained Optimization and Nonlinear equations.* Prentice-Hall, 1983.

[53] C. Djurberg, P. Svedlindh, P. Nordblad, M.F. Hansen, F. Bødker, and S. Mørup. Dynamics of an interacting particle system: Evidence of critical slowing down. *Phys. Rev. Lett.*, 79:5154–5157, 1997.

[54] J.L. Dormann and D. Fiorani, editors. *Magnetic Properties of Fine Particles.* North Holland, Amsterdam, 1992.

[55] J.L. Dormann, D. Fiorani, and E. Tronc. Magnetic relaxation in fine-particle systems. *Adv. Chem. Phys.*, 98:283–, 1997.

[56] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* John Wiley & Sons, 1998.

[57] M.-P. Dubuisson Jolly, S. Lakshmanan, and A.K. Jain. Vehicle segmentation and classification using deformable templates. *Trans. on Pattern Analysis and Machine Intelligence*, 18(3):293–308, 1996.

[58] R. D. Duda and P. E. Hart. Use of the Hough to detect lines and curves in pictures. *Communications of the ACM*, 1(15):11–15, 1972.

[59] N. Duta, A. K. Jain, and M.-P. Dubuisson-Jolly. Learning 2D shape models. In *Proc. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 8–14, 1999.

[60] N. Duta and M. Sonka. Segmentation and interpretation of mr brain images. an improved active shape model. *IEEE Transactions on Medical Imaging*, 17(6):1049–1062, 1998.

[61] G. J. Edwards, T.F. Cootes, and C. J. Taylor. Advances in active appearance models. In *Proc. Int. Conf. on Computer Vision*, pages 137–142, 1999.

[62] N. D. Efford. *Knowledge-Based Segmentation and Feature analysis of Hand Wrist Radiographs.* Tech. Report, University of Leeds, 1994.

[63] R. Fisker. *Making Deformable Template Models Operational.* PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 2000. http://www.imm.dtu.dk/documents/ftp/phdliste/phdliste.html.

[64] R. Fisker and J. M. Carstensen. Automated visual inspection of textile. In *Proc. 10th Scandinavian Conf. on Image Analysis*, pages 173–179, 1997.

[65] R. Fisker and J. M. Carstensen. On parameter estimation in deformable models. In *Proc. 14th Int. Conf. on Pattern Recognition*, pages 763–766, 1998.

[66] R. Fisker and J. M. Carstensen. A general polygon-based deformable model for object recognition. In *Proc. 8th Danish Conf. on Pattern Recognition and Image Analysis*, pages 28–35, 1999.

[67] R. Fisker, J. M. Carstensen, M.F. Hansen, F. Bødker, and S. Mørup. Estimation of nanoparticle size distributions by image analysis. *Jour. of Nanoparticle Research.* To appear.

[68] R. Fisker, J. M. Carstensen, and K. Madsen. Initialization and optimization of deformable models. In *Proc. 11th. Scandinavian Conf. on Image Analysis*, pages 295–302, 1999.

[69] R. Fisker, N. Schultz, N. Duta, and J. M. Carstensen. A general scheme for training and optimization of the Grenander deformable template model. In *Proc. Conf. on Computer Vision and Pattern Recognition*, volume I, pages 698–705, 2000.

[70] R. Fisker, N. Schultz, N. Duta, and J. M. Carstensen. The grenander deformable template model: A general scheme. 2000. Submitted.

[71] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, 1987.

[72] A. Garrido and N. Perez De La Blanca. Physically-based active shape models: Initialization and optimization. *Pattern Recognition*, 31:1003–1027, 1998.

[73] A. Garrido and N. Perez De La Blanca. Applying deformable templates for cell image segmentation. *Pattern Recognition*, 33:821–832, 2000.

[74] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 17(3):294–302, 1995.

[75] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[76] S. Geman and C.-R. Hwang. Diffusions for global optimization. *SIAM Journal on Control and Optimization*, 24(5):1031–1043, 1986.

[77] M.A. Gennert and A.L. Yuille. Determining the optimal weights in multiple objective function optimization. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 87–89, 1988.

[78] C.R. Giardina and F.P. Kuhl. Accuracy of curve approximation by harmonically related vectors with elliptical loci. *Computer Graphics and Image Processing*, 6:277–285, 1977.

[79] B. Gidas. The langevin equation as a global minimization algorithm. In *Disordered Systems and Biological Organization*, pages 321–326. Springer-Verlag, 1985.

[80] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2):155–172, 1998.

[81] M. Gleicher. Projective registration with difference decomposition. In *Proc. 1997 Conf. on Computer Vision and Pattern Recognition*, pages 331–337. IEEE Comput. Soc, 1997.

[82] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison Wesley, 1989.

[83] C. Goodall. Procrustes methods in the statistical analysis of shape. *Jour. Royal Statistical Society, Series B*, 53:285–339, 1991.

[84] G. C. Granqvist and R. A. Buhrman. Ultrafine metal particles. *J. Appl. Phys.*, 47:2200–2219, 1976.

[85] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes.* Springer, 1991.

[86] U. Grenander and M. I. Miller. Representation of knowledge in complex systems. *Jour. Royal Statistical Society, Series B*, 56(4):549–603, 1994.

[87] G.C. Hadjipanayis and R.W. Siegel, editors. *Nanophase Materials. Synthesis - Properties - Applications.* Kluwer Academic Publishers, Dordrecht, 1994.

[88] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1994.

[89] M. F. Hansen, F. Bødker, S. Mørup, C. Djurberg, and P. Svedlindh. Magnetic properties of non-interacting fe-c nanoparticles. *J. Magn. Magn. Mater.*, 928:177–181, 1998.

[90] M. F. Hansen and S. Mørup. Models for the dynamics of interacting magnetic nanoparticles. *J. Magn. Magn. Mater.*, 184:262–274, 1998.

[91] A. Hill and C.J. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision computing*, 10(5):295–300, 1992.

[92] G. Hinton, C. Williams, and M. Revow. Adaptive elastic models for handprinted character recognition. *Advances in Neural Information Processing Systems*, 4, 1992.

[93] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[94] R. Hooke and T. A. Jeeves. Direct search: solution of numerical and statistical problems. *Jour. Assoc. Comput.*, 8(212-229), 1961.

[95] P. V. C. Hough. *Method and means for recognizing complex patterns.* U.S. Patent 3069654, 1962.

[96] M. A. Hurn, I. Steinsland, and H. Rue. Parameter estimation for a deformable template model. In *Statistics Research Rep. 99.03*, Dep. of Mathematical Sciences, Uni. of Bath, 1999.

[97] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[98] J. Illingworth and J. Kitler. A survey of Hough transform. *CVGIP: Image Understanding*, 44:87–116, 1988.

[99] M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking. *Int. Jour. Computer Vision*, 29:5–28, 1998.

[100] J. Isidoro and S. Sclaroff. Active voodoo dolls: a vision based input device for nonrigid control. In *Proc. Computer Animation '98*, pages 137–143. IEEE Comput. Soc, 1998.

[101] A. K. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly. Deformable template models: A review. *Signal Processing*, 71(2):109–129, 1998.

[102] A. K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(3):267–278, 1996.

[103] A. K. Jain and D. Zonkger. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19(12):1386–1391, 1997.

[104] A.K. Jain, P.W. Duin, and M. Jianchang. Statistical pattern recognition: a review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[105] W. J. Jasper and H. Potlapalli. Image analysis of mispicks in woven fabric. *Textile Research Journal*, 65:683–692, 1995.

[106] T. Jonsson, P. Svedlindh, and M.F. Hansen. Static scaling on an interacting magnetic nanoparticle system. *Phys. Rev. Lett.*, 81:3976–3979, 1998.

[107] S. Karkanis, K. Tsoutsou, J. Vergados, and B. Dimitriadis. A real time quality inspection system for textile industries. In *Proc. of the SPIE - The Int. Society for Optical Engineering*, volume 1500, pages 164–170, Athens, Greece, 1991.

[108] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Jour. of Computer Vision*, 8(2):321–331, 1988.

[109] J. T. Kent, I. L. Dryden, and C. R. Anderson. Using circulant symmetry to model featureless objects. *Submitted*, 1999.

[110] N. Khaneja, M.I. Miller, and U. Grenander. Dynamic programming generation of curves on brain surfaces . *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 20(11):1260–1265, 1998.

[111] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *5th Int. Conf. on Computer Vision*, pages 810–815, 1995.

[112] S. Kirkpatrick, C. D. Gellant, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[113] A.K. Klein, F. Lee, and A.A. Amini. Quantitative coronary angiography with deformable spline models. *IEEE Trans. on Medical Imaging*, 16(5):468–482, 1997.

[114] A. Knoerr. *Global models of natural boundaries: Theory and applications.* PhD thesis, Brown university, Providence, Rhode Island, 1988.

[115] R. H. Kodama. Magnetic nanoparticles. *J. Magn. Magn. Mater.*, 200:359–372, 1999.

[116] F.P. Kuhl and C.R. Giardina. Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing*, 18(3):236–258, 1982.

[117] K. F. Lai and R. T. Chin. On regularization, formulation and initialization of the active contour model (snakes). In *Asian Conf. on Computer Vision*, pages 542–545, 1993.

[118] K. F. Lai and R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(11):1084–1090, 1995.

[119] S. Lakshmanan, A.K. Jain, and Y. Zhong. Detecting straight edges in millimeter-wave images. In *Proc. Int. Conf. on Image Processing*, pages 258–261, 1995.

[120] K. Lange. Convergence of EM image reconstruction algorithm with Gibbs smoothing. *IEEE Trans. on Medical Imaging*, 9(4):439–446, 1990.

[121] O. V. Larsen, P. Radeva, and E. Marti. Guidelines for choosing optimal parameters of elasticity for snakes. In *Proc. 6th Comp. Anal. of Image and Patterns*, pages 106–113, 1995.

[122] V. F. Leavers. Survey: Which Hough transform? *CVGIP: Image Understanding*, 58(2):250–264, 1993.

[123] H. Lester and S.R. Arridge. A survey of hierarchical nonlinear medical image registration. *Pattern Recognition*, 32(1):129–149, 1999.

[124] M.E. Leventon, W.E.L. Grimson, and U. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. Conf. on Computer Vision and Pattern Recognition*, volume I, pages 316–323, 2000.

[125] S. Z. Li. *Markov Random Field Modelling in Computer Vision.* Springer, 1996.

[126] S.Z. Li and Juwei Lu. Modeling bayesian estimation for deformable contours. In *Proc. Int. Conf. on Computer Vision*, pages 991–996, 1999.

[127] C.-S. Lin and C.-L. Hwang. New forms of shape invariants from elliptic fourier descriptors. *Pattern Recognition*, 20(5):535–545, 1987.

[128] P. Linares, L. Rodriguez, and G. Montilla. Genetic algorithm fitting of deformable superquadrics applied to left ventricle visualization. In *Proc. Computers in Cardiology 1998. Vol. 25 (Cat. No.98CH36292)*, volume 25, pages 657–660, 1998.

[129] J. B. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[130] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, 17(2):158–175, 1995.

[131] S. Mallat. *A wavelet tour of signal processing 2ed.* Academic Press, 1999.

[132] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 2(1):91–108, 1996.

[133] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: implementation and application to stereo. In *Proc. DARPA*, pages 720–726, 1990.

[134] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, Teller A. H., and Teller E. Equations of state calculations by fast computing machines. *Jour. Chemical Physics*, 21:1087–1092, 1953.

[135] K. Miettinen. *Evolutionary algorithms in engineering and computer science : Recent advances in genetic algorithms.* Winston and Wiley, 1999.

[136] S. Mitchell, B. Lelieveldt, R. Geest, J. Schaap, J. Reiber, and M. Sonka. Segmentation of cardiac mr images: An active appearance model approach. In *Medical Imaging 2000: Image Processing, San Diego CA, SPIE*, volume 1. SPIE, 2000.

[137] D. Mumford. *The Bayesian Rational for Energy Functionals.* Kluwer Academics, Dordrecht, 1994.

[138] C. Neubauer. Segmentation of defects in textile fabric. In *Proc. 11th Int. Conf. on Pattern Recognition 1992*, volume I, pages 688–691, 1991.

[139] A. Neumann and C. Lorenz. Statistical shape model based segmentation of medical images. *Computerized Medical Imaging and Graphics*, 22(2):133–143, 1998.

[140] L. Norton-Wayne. The automated inspection of moving webs using machine vision. In *Proc. IEE Colloquium 'The Application of Machine Vision*, volume 3, pages 1–8, London, UK, 1995.

[141] K. O'Grady and A. Bradbury. Particle size analysis in ferrofluids. *J. Magn. Magn. Mater.*, 39:91–94, 1983.

[142] S. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *Jour. Computational Physics.*, 79(4):12–49, 1988.

[143] N. Paragios and R. Deriche. Erratum - corrections to "geodesic active contours and level sets for the detection and tracking of moving objects". *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 22(4):415, 2000.

[144] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 22(3):266–280, 2000.

[145] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 21(6):564–569, 1999.

[146] N. Peterfreund. The velocity snake: deformable contour for tracking in spatio-velocity space. *Computer Vision and Image Understanding*, 73(3):346–356, 1999.

[147] T. Poggio and V. Torre. Ill-posed problems and regularization analysis in early vision. In *Proc. AARPA Image understaading workshope*, pages 257–263, 1984.

[148] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985.

[149] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[150] J. A. Rice. *Mathematical Statistics and Data Analysis, 2. Edition.* Duxbury Press, 1995.

[151] B.D. Ripley and A.I. Sutherland. Finding spiral structures in images of galaxies. *Philosophical Trans. of the Royal Society, Series A*, 332(1627):477–485, 1990.

[152] Y. Rubner and C. Tomasi. Texture metrics. In *Int. Conf. on Systems, Man, and Cybernetics*, volume 5, pages 4601–5607, 1998.

[153] *SAS Procedures Guide, Ver. 6, Third ed.* SAS Institute Inc., Cary, NC, USA, 1990.

[154] N. Schultz. *Segmentation and Classification of Biological Objects.* PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 1995.

[155] N. Schultz and K. Conradsen. 2D vector cycle deformable templates. *Signal Processing*, 71(2):141–153, 1998.

[156] S. Sclaroff and J. Isidoro. Active blobs. *Proc. 6th Int. Conf. on Computer Vision*, pages 1146–1153, 1998.

[157] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):545–61, 1995.

[158] G. Scott. The alternative snake - and other animals. In *Proc. 3rd Alvey Vision Conf.*, pages 341–347, 1987.

[159] J. A. Sethian. *Level Set Methods Fast Marching 2ed.* Cambrige University Press, 1999.

[160] B. Shahraray and D. J. Anderson. Optimal estimation of contour properties by cross-validated regularization. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 11(6):600–610, 1989.

[161] J.R. Shewchuk. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry. FCRC'96 Workshop.*, pages 203–222. Springer-Verlag, 1996.

[162] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision.* Chapman & Hall, 1993.

[163] L. H. Staib and S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 14(11):1061–1075, 1992.

[164] M. B. Stegmann. Active appearance models: Theory, extensions and cases. Master's thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 2000.

[165] M. B. Stegmann, R. Fisker, and Ersbøll B. K. Extending and applying active appearance models for automated, high precision segmentation in different image modalities. 2000. Submitted.

[166] G. C. Stockman and A. K. Agrawala. Equivalence of Hough curve detection and to template matching. *Communications of the ACM*, 1(20):820–822, 1977.

[167] G. Storvik. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 16(10):976–986, 1994.

[168] G. Strang and T. Nguyen. *Wavelets and filter banks.* Wellesley-Cambridge Press, Feb. 1986.

[169] T. Sugimoto, Y. Wang, H. Itoh, and A. Muramatsu. Systematic control of size, shape and internal structure of monodisperse α-fe$_2$o$_3$ particles. *Colloid Surfaces A: Physicochem. Eng. Aspects*, 134:265–279, 1998.

[170] H. D. Tagare. Deformable 2-d template matching using orthogonal curves. *IEEE Trans. on Medical Imaging*, 16(1):108–177, 1997.

[171] D Terzopoulos and R. Szeliski. Tracking with kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. MIT Press, 1992.

[172] T. Thomas. Vision system applied to the quality control in the textile industry. In *Proc. Int. Conf. on Quality Control by Artificial Vision*, volume 3, pages 308–319, Le Creusot, France, 1995.

[173] A. N. Tikhonov. Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4:1624–1627, 1963.

[174] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 586–91, 1991.

[175] M. Unser, A. Aldroubi, and M. Eden. On the asymptotic convergence of b-spline wavelets to gabor functions. *IEEE Trans. Information Theory*, 38(2):864–872, 1992.

[176] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms . *Signal Processing*, 30(2):141–162, 1993.

[177] P. A. van den Elsen, E-J. D. Pol, and M. A. Viergever. Medical image matching - a review with classification. *IEEE Eng. Med. Biol*, 12:26–39, 1993.

[178] J. van Wonterghem, S. Mørup, S.W. Charles, and S. Wells. An investigation of the chemical reaction leading to the formation of ultrafine amorphous $\mathrm{fe}_{100-x}\mathrm{c}_x$ alloy particles. *J. Colloid Interface Sci.*, 121:558–563, 1988.

[179] J. van Wonterghem, S. Mørup, S.W. Charles, S. Wells, and J. Villadsen. Formation of a metallic glass by thermal decomposition of $\mathrm{fe(co)}_5$. *Phys. Rev. Lett.*, 55:410–413, 1985.

[180] G. Wahba. *Spline models for observation data.* Soc. for Industrial and Applied Math., 1990.

[181] G. Wahba and S. Wold. Periodic splines for spectral density estimation. the use of cross validation for determining the degree of smoothing. *Commun. Stat.*, 4(2):125–141, 1975.

[182] K.N. Walker, T. F. Cootes, and C. J. Taylor. Determining correspondences for statistical models of facial appearance. In *Proc. Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 271–276. IEEE Comput. Soc, 2000.

[183] J. West et al. Comparison and evaluation of retrospective intermodality brain image registration techniques. *Jour. of Computer Assisted Tomography*, 21(4):554–566, 1997.

[184] C.B.H. Wolstenholme and C.J. Taylor. Wavelet compression of active appearance models. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99*, pages 544–554, 1999.

[185] C. F. J. Wu. On the convergence properties of the EM algortihm. *The Annals of Statistics*, 11:95–103, 1983.

[186] Jr. Yezzi, A., A. Tsai, and A. Willsky. Fully global coupled curves evolution equations for image segmentation. Technical report, LIDS, January 1999. Submitted to PAMI.

[187] Jr. Yezzi, A., A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proc. Int. Conf. on Computer Vision*, pages 898–903, 1999.

[188] H. Yoshida, S. Katsuragawa, Y. Amit, and Kunio Doi. Wavelet-based deformable contour and its application to detection of pulmonary nodules on chest radiographs. In *Proc. of the SPIE*, volume 3169, pages 328–336, 1997.

[189] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *Int. Jour. of Computer Vision*, 8(2):99–111, 1992.

[190] Y. Zhong. *Object Matching Using Deformable Templates.* PhD thesis, Department of Computer Science, Michigan State University, MI 48823, USA, 1997.

[191] Y. Zhong and A.K. Jain. Object localization using color, texture and shape. *Pattern Recognition*, 33(4):671–684, 2000.

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences. xiv* + 143 pp.

2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems. xiii* + 232 pp.

3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns. xv* + 136 pp.

4. **Melgaard, Henrik.** (1994). *Identification of physical models. xvii* + 246 pp.

5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system. xxii* + 153 pp.

6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data. xxiv* + 213 pp.

7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications. xviii* + 345 pp.

8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxxviii* + 282 pp., 283-569 pp.

9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer. viii* + 174 pp.

10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).

11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions. iv* + 154 pp.

12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems. xiv* + 348 pp.

13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects. xiv* + 194 pp.

14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems. xiv* + 120 pp.

15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations. iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems. xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation. xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods. xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics. viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points. xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics. xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surgery simulation. xxvii* + 274 pp.

26. **Bendtsen, Claus.** (1996). *Parallel numerical algorithms for the solution of systems of ordinary differential equations. viii* + 79 pp.

27. **Lauritsen, Morten Bach.** (1997). *Delta-domain predictive control and identification for control. xxii* + 292 pp.

28. **Bischoff, Svend.** (1997). *Modelling colliding-pulse mode-locked semiconductor lasers. xxii* + 217 pp.

29. **Arnbjerg-Nielsen, Karsten.** (1997). *Statistical analysis of urban hydrology with special emphasis on rainfall modelling.* Institut for Miljøteknik, DTU. *xiv* + 161 pp.

30. **Jacobsen, Judith L.** (1997). *Dynamic modelling of processes in rivers affected by precipitation runoff. xix* + 213 pp.

31. **Sommer, Helle Mølgaard.** (1997). *Variability in microbiological degradation experiments - Analysis and case study. xiv* + 211 pp.

32. **Ma, Xin.** (1997). *Adaptive extremum control and wind turbine control. xix* + 293 pp.

33. **Rasmussen, Kim Ørskov.** (1997). *Nonlinear and stochastic dynamics of coherent structures. x* + 215 pp.

34. **Hansen, Lars Henrik.** (1997). *Stochastic modelling of central heating systems. xxii* + 301 pp.

35. **Jørgensen, Claus.** (1997). *Driftsoptimering på kraftvarmesystemer.* 290 pp.

36. **Stauning, Ole.** (1997). *Automatic validation of numerical solutions. viii* + 116 pp.

37. **Pedersen, Morten With.** (1997). *Optimization of recurrent neural networks for time series modeling. x* + 322 pp.

38. **Thorsen, Rune.** (1997). *Restoration of hand function in tetraplegics using myoelectrically controlled functional electrical stimulation of the controlling muscle. x* + 154 pp. + Appendix.

39. **Rosholm, Anders.** (1997). *Statistical methods for segmentation and classification of images. xvi* + 183 pp.

40. **Petersen, Kim Tilgaard.** (1997). *Estimation of speech quality in telecommunication systems. x* + 259 pp.

41. **Jensen, Carsten Nordstrøm.** (1997). *Nonlinear systems with discrete and continuous elements.* 195 pp.

42. **Hansen, Peter S.K.** (1997). *Signal subspace methods for speech enhancement. x* + 226 pp.

43. **Nielsen, Ole Møller.** (1998). *Wavelets in scientific computing. xiv* + 232 pp.

44. **Kjems, Ulrik.** (1998). *Bayesian signal processing and interpretation of brain scans. iv* + 129 pp.

45. **Hansen, Michael Pilegaard.** (1998). *Metaheuristics for multiple objective combinatorial optimization. x* + 163 pp.

46. **Riis, Søren Kamaric.** (1998). *Hidden markov models and neural networks for speech recognition. x* + 223 pp.

47. **Mørch, Niels Jacob Sand.** (1998). *A multivariate approach to functional neuro modeling. xvi* + 147 pp.

48. **Frydendal, Ib.** (1998.) *Quality inspection of sugar beets using vision. iv* + 97 pp. + app.

49. **Lundin, Lars Kristian.** (1998). *Parallel computation of rotating flows. viii* + 106 pp.

50. **Borges, Pedro.** (1998). *Multicriteria planning and optimization. - Heuristic approaches. xiv* + 219 pp.

51. **Nielsen, Jakob Birkedal.** (1998). *New developments in the theory of wheel/rail contact mechanics. xviii* + 223 pp.

52. **Fog, Torben.** (1998). *Condition monitoring and fault diagnosis in marine diesel engines. xii* + 178 pp.

53. **Knudsen, Ole.** (1998). *Industrial vision. xii* + 129 pp.

54. **Andersen, Jens Strodl.** (1998). *Statistical analysis of biotests. - Applied to complex polluted samples. xx* + 207 pp.

55. **Philipsen, Peter Alshede.** (1998). *Reconstruction and restoration of PET images. vi* + 132 pp.

56. **Thygesen, Uffe Høgsbro.** (1998). *Robust performance and dissipation of stochastic control systems.* 185 pp.

57. **Hintz-Madsen, Mads.** (1998). *A probabilistic framework for classification of dermatoscopic images. xi* + 153 pp.

58. **Schramm-Nielsen, Karina.** (1998). *Environmental reference materials methods and case studies. xxvi* + 261 pp.

59. **Skyggebjerg, Ole.** (1999). *Acquisition and analysis of complex dynamic intra- and intercellular signaling events.* 83 pp.

60. **Jensen, Kåre Jean.** (1999). *Signal processing for distribution network monitoring. xv* + 199 pp.

61. **Folm-Hansen, Jørgen.** (1999). *On chromatic and geometrical calibration. xiv* + 238 pp.

62. **Larsen, Jesper.** (1999). *Parallelization of the vehicle routing problem with time windows.xx* + 266 pp.

63. **Clausen, Carl Balslev.** (1999). *Spatial solitons in quasi-phase matched struktures. vi* + (flere pag.)

64. **Kvist, Trine.** (1999). *Statistical modelling of fish stocks. xiv* + 173 pp.

65. **Andresen, Per Rønsholt.** (1999). *Surface-bounded growth modeling applied to human mandibles. xxii* + 125 pp.

66. **Sørensen, Per Settergren.** (1999). *Spatial distribution maps for benthic communities.*

67. **Andersen, Helle.** (1999). *Statistical models for standardized toxicity studies. viii* + (flere pag.)

68. **Andersen, Lars Nonboe.** (1999). *Signal processing in the dolphin sonar system. xii* + 214 pp.

69. **Bechmann, Henrik.** (1999). *Modelling of wastewater systems. xviii* + 161 pp.

70. **Nielsen, Henrik Aalborg.** (1999). *Parametric and non-parametric system modelling. xviii* + 209 pp.

71. **Gramkow, Claus.** (1999). *2D and 3D object measurement for control and quality assurance in the industry. xxvi* + 236 pp.

72. **Nielsen, Jan Nygaard.** (1999). *Stochastic modelling of dynamic systems. xvi* + 225 pp.

73. **Larsen, Allan.** (2000). *The dynamic vehicle routing problem. xvi* + 185 pp.

74. **Halkjær, Søren.** (2000). *Elastic wave propagation in anisotropic inhomogeneous materials. xiv* + 133 pp.

75. **Larsen, Theis Leth.** (2000). *Phosphorus diffusion in float zone silicon crystal growth. viii* + 119 pp.

76. **Dirscherl, Kai.** (2000). *Online correction of scanning probe microscopes with pixel accuracy.* 146 pp.

77. **Fisker, Rune.** (2000). *Making deformable template models operational. xx* 220 pp.

78. **Hultberg, Tim Helge.** (2000).*Topics in computational linear optimization. xiv* + 180 pp.

79. **Andersen, Klaus Kaae.**(2000).*Stochastic modelling of energy systems. xiv* + 191 pp.

80. **Thyregod, Peter.** (2001). *Modelling and monitoring in injection molding. xvi* + 132 pp.

Index