A general scheme for training and optimization of the Grenander deformable template model

Fisker, Rune; Schultz, Nette; Duta, N.; Carstensen, Jens Michael

Published in:
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition

Link to article, DOI: 10.1109/CVPR.2000.855888

Publication date:
2000

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
A General Scheme for Training and Optimization of the Grenander Deformable Template Model

Rune Fisker\textsuperscript{a}, Nette Schultz, Nicolae Dută\textsuperscript{b} and Jens Michael Carstensen\textsuperscript{a}

\textsuperscript{a}Department of Mathematical Modelling, Technical University of Denmark, 2800, Denmark
\textsuperscript{b}Department of Computer Science, Michigan State University, MI 48823, USA
rf@imm.dtu.dk

Abstract

General deformable models have reduced the need for hand-crafting new models for every new problem. But still most of the general models 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. \cite{7, 13} to a new problem with minimal manual interaction, beside supplying a training set, which can be done by a non-expert user. The main contributions compared to previous work are a supervised learning scheme for the model parameters, a very fast general initialization algorithm and an adaptive likelihood model based on local means. The model parameters are trained by a combination of a 2D shape learning algorithm and a Maximum Likelihood based criteria. The fast initialization algorithm is based on a search approach using a filter interpretation of the likelihood model.

Index terms: deformable templates, active contour models, initialization, model parameter estimation.

1 Introduction

The deformable template model literature is very rich in different models, see \cite{1, 10, 15} for a survey of the best known models. One reason for this is a general tendency to hand craft a new model for every new problem, even though existing general models, such as \cite{1, 2, 4, 7, 11, 17}, 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, \( \theta \), 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 a small training set, and eventually run an incremental or iterative learning algorithm such as the Expectation-Maximization. No matter whether a training set is created manually or by some iterative algorithm, a method for estimation of the model parameters, \( \theta \), based on a training set and a general initialization/optimization method is needed. In this paper we proposed such methods for the general deformable template model proposed by Ulf Grenander et al. \cite{7, 13} and further investigated and developed in \cite{8, 9, 12, 16}.

2 The Grenander model.

The Grenander Model is formulated in the Baysian framework. To make inference the posterior distribution \( P(v|y, \theta) \) is maximized, where \( v \) is the template parameters defining the object, \( y \) is the image and \( \theta \) is the model parameters. Using Bayes theorem \( P(v|y, \theta) \propto P(y|v, \theta)P(v|\theta) \) the posterior can be separated into the prior \( P(v|\theta) \) and the likelihood \( P(y|v, \theta) \). In the Grenander Model the object is represented by a set of vertices \( v = (v_1^T, ..., v_n^T)^T \in \Omega_n \subset \mathbb{R}^{2n} \), where \( v_i = (x_i, y_i)^T \). From the vertices \( v \) the corresponding edges \( z \in \mathbb{R}^{2n} \) can be calculated, where \( z_i = v_i - v_{i-1} \) in a cyclic manner (see figure 1).

The template vector cycle \( z = (z_0^T, ..., z_{n-1}^T)^T \in \mathbb{R}^{2n} \) must satisfy the closure constraint: \( \sum_{i=0}^{n-1} z_i^{(x)} = 0 \) and \( \sum_{i=0}^{n-1} z_i^{(y)} = 0 \), where \( z_i = (z_i^{(x)}, z_i^{(y)}) \). In general an object can either be represented by the vertices \( v \) or by the edges \( z \) and an offset \( v_{n-1} \). The relationship is:

\[
v = Ev_{n-1} + Fz
\]  
(1)
2.1 Prior model.

Assume that a mean or prototypal shape \( \bar{\varphi} = (\bar{v}_i^0, ..., \bar{v}_{n-1}^i)^T \in \Omega_v \) exists. Given a new shape \( \varphi \) with the same number of vertices, the corresponding change in local scale and orientation of each mean edge \( z_i = S_i(\kappa_i, \phi_i)\bar{z}_i \) can be determined. Matrices changing scale and orientation are:

\[
E = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 \\
0 & 1 \\
1 & 0 \\
\vdots & \\
\end{bmatrix}
\]

and

\[
F = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \ldots \\
0 & 1 & 0 & 0 & 0 & \ldots \\
1 & 0 & 1 & 0 & 0 & \ldots \\
0 & 1 & 0 & 1 & 0 & \ldots \\
1 & 0 & 1 & 0 & 1 & \ldots \\
\vdots & & & & & \ddots
\end{bmatrix}
\]

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

\[
S_i(\kappa_i, \phi_i) = \begin{bmatrix}
1 + \kappa_i & -\phi_i \\
\phi_i & 1 + \kappa_i
\end{bmatrix}
\]

where \( \kappa_i = \gamma \cos(\varphi) - 1 \) and \( \phi_i = -\gamma \sin(\varphi) \). For small values of \( \varphi \) and \( \gamma \), near 1, \( \kappa_i \approx \gamma \) and \( \phi_i \approx -\varphi \) 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 \( \kappa = (\kappa_0, ..., \kappa_{n-1}) \) and \( \phi = (\phi_0, ..., \phi_{n-1}) \) are assumed to follow an independent cyclic first-order Gaussian-Markov process with mean zero [7]. The density of a first-order Gaussian-Markov process with mean zero for the n-cyclic parameter vector \( \kappa \) is defined by:

\[
f(\kappa) = \frac{1}{\sqrt{2\pi} \sqrt{|R_n|}} \exp\left\{ -\frac{1}{2} \kappa R_n^{-1} \kappa \right\}
\]

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

\[
R_n^{-1} = \begin{bmatrix}
\alpha_0 + 2\alpha_1 & -\alpha_1 & 0 & \ldots & -\alpha_1 \\
-\alpha_1 & \alpha_0 + 2\alpha_1 & -\alpha_1 & \ldots & 0 \\
0 & -\alpha_1 & \alpha_0 + 2\alpha_1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
-\alpha_1 & 0 & 0 & \ldots & \alpha_0 + 2\alpha_1
\end{bmatrix}
\]

\( \alpha = (\alpha_0, \alpha_1) \) are such that \( R_n^{-1} \) is positive definite.

This parameterization gives an more intuitive interpretation of the parameters controlling \( R_n^{-1} \) than the original parameterization in [7]. The parameters \( \alpha \) are easier to interpret in the following form of the density (5):

\[
f(\kappa) = \frac{1}{c} \exp\left\{ -\frac{1}{2} \sum_{i=0}^{n-1} (\alpha_0 (\kappa_i)^2 + \alpha_1 (\kappa_i - \kappa_{i-1})^2) \right\}
\]

where \( c = 2\pi^{n/2} |R_n|^{1/2} \). From (7) it can be seen that \( \alpha_0 \) controls the likelihood of how much \( \kappa_i \) differs from zero. For large values of \( \alpha_0 \) it is very unlikely that \( \kappa_i \) differs very much from zero, because for large values of \( \alpha_0 \), \( f(\kappa) \approx 0 \) even for small values of \( \kappa_i \). \( \alpha_1 \) controls the neighbor relationship between \( \kappa_i \) and \( \kappa_{i+1} \). For positive values of \( \alpha_1 \) neighbors tend to be positively correlated and the opposite for negative. Large values of \( \alpha_1 \) correspond to tight bonding between neighbors.

An equivalent density distribution is derived for the orientation parameter vector \( \phi \) with weight parameters \( \alpha_0 \) and \( \alpha_1 \). In the original formulation individual weights, \( \alpha_0^i, \alpha_1^i, \alpha_0^i, \alpha_1^i \), are assigned to each \( \kappa_i \) and \( \phi_i \), but when the model is actually applied all weights are assumed to be equal \( \alpha_0^i = \alpha_1^i = \alpha_0 \) and \( \alpha_1^i = \alpha_1^i = \alpha_1 \), see [8, 12, 16]. To simplify the notation the index is omitted initially. Recall \( \bar{z}_i = S_i(\kappa_i, \phi_i)\bar{z}_i \). This can be rearranged into:

\[
z_i = \begin{bmatrix}
\frac{\bar{z}_x}{\bar{z}_y} \\
\frac{\bar{z}_y}{\bar{z}_x}
\end{bmatrix}
\begin{bmatrix}
\kappa_i \\
\phi_i
\end{bmatrix} + \bar{z}_i
\]

Rearranging the global vector cycle this way gives:

\[
z = G \omega + \bar{z} \quad \text{for} \quad G = \Gamma H
\]
where $\omega = (\kappa^T, \phi^T)^T$,
\[
\Gamma = \begin{bmatrix}
\frac{z_0}{\bar{z}_0} & -\frac{y_0}{\bar{z}_0} & 0 & 0 & \cdots \\
\frac{z_0}{\bar{z}_0} & \frac{y_0}{\bar{z}_0} & 0 & 0 & \cdots \\
0 & 0 & \frac{z_0}{\bar{z}_0} & -\frac{y_0}{\bar{z}_0} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

(10)

and $H$ is a permutation matrix interchanging rows:
\[
H = \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \cdots & 1 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0
\end{bmatrix}
\]

(11)

If $\kappa \sim N(0, R_{\kappa})$ and $\phi \sim N(0, R_{\phi})$ are stochastically independent, the following distribution is obtained from (9) for the edges $z \sim N(\tilde{z}, R_z)$ where
\[
R_z = GR_zG^T \text{ for } R_{\omega} = \begin{bmatrix}
R_{\kappa} & 0 \\
0 & R_{\phi}
\end{bmatrix}
\]

(12)

Unfortunately this distribution $z \sim N(\tilde{z}, R_z)$ does not satisfy the closure constraint, i.e. a sample drawn from $N(\tilde{z}, R_z)$ will in general not be closed. The closure constraint can be rewritten to $Lz = 0$, where
\[
L = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & \cdots \\
0 & 1 & 0 & 1 & 0 & 1 & \cdots
\end{bmatrix} \in \mathbb{R}^{2 \times n}
\]

(13)

Using the theory on conditioning Gaussian distributions the template vector distribution conditioning on closure is $[z | Lz = 0] \sim N(\tilde{z}, R)$ where
\[
R = R_z - R_z L(LR_zL^T)^{-1} LR_z
\]

(14)

The final distribution of the vertices, $v$, is:
\[
v \sim N(Ev_{n-1} + F\tilde{z}, FRF^T)
\]

(15)

The prior model $P(v | \tilde{v}, \alpha)$ is identical to the density function of $N(Ev_{n-1} + F\tilde{z}, FRF^T)$.

2.2 Likelihood model with global mean.

The original likelihood (or observation) model $P(v | y, \theta)$ is based on the assumption that the pixel values, $y_{r,c}$, inside and outside the object are independently Gaussian distributed with mean $\mu_{in}$ and $\mu_{out}$ and common variance $\lambda$. This assumption leads to the following likelihood:
\[
P(v | y, \theta) = \prod_{(r,c) \in \Omega_{in}} \frac{1}{\sqrt{2\pi} \lambda} \exp\left\{-\frac{(y_{r,c} - \mu_{in})^2}{2\lambda}\right\} \times 
\prod_{(r,c) \in \Omega_{out}} \frac{1}{\sqrt{2\pi} \lambda} \exp\left\{-\frac{(y_{r,c} - \mu_{out})^2}{2\lambda}\right\}
\]

(16)

where $\Omega_{in}$ and $\Omega_{out}$ are the set of pixels, which are inside and outside the template, 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 $\rho_u$ and horizontal $\rho_h$ inflation of the smallest rectangle, which contains the actual initial configuration.

3 Model parameter estimation.

To actually apply the Grenander model values need to be assigned to the model parameters $\theta = (\tilde{v}, \alpha, \mu_{in}, \mu_{out}, \lambda, \rho_u, \rho_h)$. In the following sections a fully automated scheme for estimation of the model parameters based on m manually marked object outlines, $o_0, \ldots, o_{m-1}$, in the images, $y_0, \ldots, y_{m-1}$, is presented. Let the outlines be without point correspondence and let the number of points be much larger than the final number of vertices $n$. Compared to a point-based training set with established point correspondence, the outlines are more robust and easier to create, primarily because points do not have to be placed at corresponding positions on each object. Beside the outlines the rectangles, $a_{out,0}, \ldots, a_{out,m-1}$, defining the local areas need to be given.

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

3.1 Prior mean shape.

The crucial step in the estimation of the mean template is to create the aligned parameterization $v'_0, \ldots, v'_{m-1}$ of the outlines $o_0, \ldots, o_{m-1}$. Note that $v'_0, \ldots, v'_{m-1}$ are aligned with respect to the Euclidean transformation to remove the random Euclidean changes. The mean template $\tilde{v}'$ is then calculated as the simple average of the vertices $v'_0, \ldots, v'_{m-1}$.

The parameterization implies a subsampling and a registration process. A 2D Shape Learning algorithm recently proposed by Dutta et al. [3] 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.

One of the outputs from the algorithm is the aligned templates with established point correspondences. The templates are aligned to the last template
by minimizing the Procrustes distance. This does not
insure that the average of the relative size and rotation
is one and zero, respectively, which it is supposed to be
if the mean template $\bar{v}_i$ should be a mean with respect
to size and orientation and not just shape. These
properties are obtained by scaling and rotating the output
from the shape learning algorithm by $\frac{1}{m} \sum_{i=0}^{m-1} \frac{1}{n}$ and
$-\frac{1}{m} \sum_{i=0}^{m-1} \varphi_i$, where $\gamma_i$ and $\varphi_i$ are the scale and rotation,
which aligned shape $i$ with the last shape.

3.2 Prior weight parameters

Most deformable models contain eight parameters, but never less than eight 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. [14] is one exception.
They use a non-linear minimax criteria to estimate the
regularization parameter. Fisker & Carstensen [8]
uses a combined maximum likelihood and minimum
distance criteria to estimate eight parameters in a
deformable model used for textile inspection. Recent
work by Kent et al. [12] and Hurn et al. [9] has
considered the question of weight parameter estimation
in Geman related models. The work by Kent et al.
[12] only considers circular symmetry objects and
the work by Hurn et al. [9] uses a closure constraint
based on $v_0 = v_n$. 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 (15),
which has an almost homogeneous covariance structure.

The estimation of the weight parameters are based
on the aligned training samples, $v_0^t, \ldots, v_{m-1}^t$. Using
the assumption that $v_i^t$ is a sample from the prior distribution $N(Ev_{m-1} + \tilde{F}z, FRT^T)$, we propose to estimate
the weights $\alpha$ using the Maximum Likelihood
(ML) estimator:

$$\hat{\alpha} = \max_\alpha \prod_{i=0}^{m-1} P(v_i^t | \alpha)$$

(17)

To evaluate $P(v_i^t | \alpha)$ the inverse of $FRT^T$ need
to be calculated. Unfortunately this is not straightforward,
because $R$ is rank deficient, due the closure constraint,
$\sum_{i=0}^{m-1} z_i = 0$, which removes two degrees of freedom, i.e., rank($FRT^T$) = $2n - 2$. This problem
is solved by the use of the full weight pseudo inverse
of $FRT^T$. Due to the closure constraint the last two rows and columns of $FRT^T$ will always be zero. This
is basically also the reason for the last element of $\alpha
being identical to the offset $v_{m-1}$ in (1). Based on this
observation the pseudo inverse $(FRT^T)^-$ is defined by:

$$(FRT^T)^- = \begin{bmatrix}
\Sigma_{11}^{-1} & 0 \\
0 & 0
\end{bmatrix}$$

(18)

where $\Sigma_{11}$ is the first $2n - 2$ rows and columns of $FRT^T$.

3.3 Likelihood parameters

The means, $\mu_{in}, \mu_{out}$, and the common variance,
$\lambda$, used in the likelihood model can be estimated from
$v_0^t, \ldots, v_{m-1}^t, a_{out,0}, \ldots, a_{out,m-1}$ and the corresponding
images, $y_0, \ldots, y_{m-1}$, by calculating the traditional
means and the pooled variance of the pixels values
inside and outside the template.

The true vertical $p_v$ and horizontal $p_h$ inflation cannot
be estimated directly, so $p_v$ and $p_h$ are approximated
by the average vertical and horizontal inflations,
which map the smallest rectangle containing $v_i^t
\leftrightarrow a_{out,i}$ independent of position. This approximation is
based on the assumption, that the initial configuration
will be placed close to the center of $a_{out,i}$ with
approximately the same size as $v_i$. 

4 Initialization

To actually make inference about an object in an
image $y$, estimation of the maximum a posteriori
(MAP), $\hat{\theta} = \max_\theta P(\theta | y, \theta)$, 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
[17], the generalized Hough transform [6, 14] and
moment [1] have been applied with success to the respec-
tive models, but these approaches cannot be applied in
the general setting of the Geman model. The only
general initialization approach, which haveknowledge
about, is the search strategy [4, 7, 11], where a
sparse search is performed in the parameter space $\Omega,
using either a deterministic [4, 11] or randomized [7]
strategy. A disadvantage of the randomized search
strategy used in [7] is that the position should be
known well, if the model should not be caught
in a local minimum. For the general setting this makes
the deterministic search more attractive.

For most problems the initial position is unknown,
and a computational expensive part of the determin-
istic search is to shift different configurations of the
template around the image and calculate the posterior
probability $P(\theta | y, \theta)$. Based on this observation
we proposed a fast search strategy using a filter interpre-
tation $f_k(v, \theta)$ of the likelihood energy $U(y | v, \theta)$,
where \( P(y|v, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mu)^2\right) \), 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 \( f_i(v, \theta) \) and the image \( y \). The likelihood energy \( U(y|v, \theta) \), derived directly from \( P(y|v, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mu)^2\right) \) in (16), can be rearranging into a filtering in the following way (note this is still only for one position of the template):

\[
U(y|v, \theta) = \sum_{\Omega_{\text{in}}} \frac{(y - \mu_{\text{in}})^2}{2\lambda} + \sum_{\Omega_{\text{out}}} \frac{(y - \mu_{\text{out}})^2}{2\lambda} - \sum_{\Omega_{\text{in}}} \frac{y^2}{\lambda} - \sum_{\Omega_{\text{out}}} y\mu_{\text{out}} - \frac{\Omega_{\text{in}}}{2\lambda}\mu_{\text{in}}^2 + \frac{\Omega_{\text{out}}}{2\lambda}\mu_{\text{out}}^2 - \frac{y^2}{2\lambda} + \frac{\Omega_{\text{in}}}{2\lambda}\mu_{\text{in}}^2 + \frac{\Omega_{\text{out}}}{2\lambda}\mu_{\text{out}}^2 - y * f_i(v, \mu_{\text{in}}, \mu_{\text{out}}, \lambda)
\]

where \( |\Omega| \) is the number of pixels in \( \Omega \), \( \sum_{\Omega} y \) is a short form for \( \sum_{(r,c)\in \Omega} y_{r,c} \), \( y * f_i(v, \mu_{\text{in}}, \mu_{\text{out}}, \lambda) \) is the convolution for one position and \( f_i(v, \mu_{\text{in}}, \mu_{\text{out}}, \lambda) \) is the likelihood interpreted filter, which has the size of \( a_{\text{out}} \) and contain \( \mu_{\text{in}} / \lambda \) and \( \mu_{\text{out}} / \lambda \) 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 \( y \) and \( f_i(v, \mu_{\text{in}}, \mu_{\text{out}}, \lambda) \) is performed for the ROI and the terms \( \sum_{\Omega_{\text{in}}} \frac{y^2}{\lambda} \) and \( \sum_{\Omega_{\text{out}}} \frac{\mu_{\text{in}}^2}{2\lambda} + \frac{\mu_{\text{out}}^2}{2\lambda} \) 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 the removed need for recalculations at each position. Another significant gain in computation time can be obtained, if the image can be resampled/zoomed into having \( 2^s \) columns and \( 2^s \) rows, respectively, where \( s < N_\text{in} \) and \( s < N_\text{in} \). 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:

1. Create relevant template configurations \( v_0, ..., v_{k-1} \).
2. Create likelihood interpreted filters \( f_i(v_i, \theta) \) (and optionally their Fourier transform \( F_i(v_i, \theta) \)).
3. Calculate \( P(y|v_i, \theta) \) for the center of the template corresponding to each pixel within the ROI by a correlation of \( f_i(v_i, \theta) \) and \( y \).
4. Calculate \( P(v_i|y, \theta) = P(v_i|\theta)P(y|v_i, \theta) \).
5. \( i = i + 1. \) Go to 3 if \( i < k \).
6. Extract the initial configurations from the calculated values of \( P(v|y, \theta) \).

The actual choice and number of initial configurations \( v_0, ..., v_{k-1} \) 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. In the normal setting the ROI is chosen, such that the full template is always inside the image.

The final step is to extract the initial configurations from the calculated \( P(v|y, \theta) \). In practice this is done by extracting the maximum posterior \( P(v|y, \theta) \mid (r,c) \) 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 object in the image are known, the initial configurations are extracted as the corresponding number of local maxima with the highest posterior probability \( P(v|y, \theta) \mid (r,c) \). In the case where the number of objects are unknown the initial configurations are extracted as local maxima with a posterior probability \( P(v|y, \theta) \mid (r,c) \) above a threshold \( t_p \).

5 Optimization

The optimization is performed using a Simulated Annealing scheme incorporating the Metropolis algorithm. The temperature is decreased by an exponential temperature scheme \( T_{i+1} = k_T T_i \), and the new sample \( v_{\text{new}} \) is generated by sampling in the prior distribution \( P(v|\theta) \). In practice only a small segment \( v_{\text{seg}} = (v_0, ..., v_{i+k}, ..., v_{i+k}) \) of the vector cycle is changed at each iteration. The samples are then generated from the conditional prior distribution \( P(v|v_{\text{seg}}, \theta) \), where \( v_{\text{seg}} = (v_0, ..., v_{i-1}, v_{i+k}, ..., v_{n-1}) \). Due to the first-order Markov assumption the conditional distribution only depends on its neighbors \( P(v_{\text{seg}}|v_{i+k}, \theta) = P(v_{\text{seg}}|v_{i-1}, v_{i+k}, \theta) \). In practice the conditional distribution \( P(v_{\text{seg}}|v_{i-1}, v_{i+k}, \theta) \) is obtained from the conditional distributions \( f(\kappa_{\text{seg}}|\kappa_{i-1}, \kappa_{i+k}) \) and \( f(\phi_{\text{seg}}|\phi_{i-1}, \phi_{i+k}) \), which are propagated through a similar framework as used for the full prior distribution in section 2.1. Refer to [16] for the derivation.

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

\[
\frac{P(y|v_{\text{new}}, \theta)}{P(y|v_{\text{old}}, \theta)} = \exp\left\{ \sum_{\Omega_{\text{in}}^{\text{new}}} \frac{(y - \mu_{\text{in}})^2}{2\lambda} - \sum_{\Omega_{\text{out}}^{\text{new}}} \frac{(y - \mu_{\text{out}})^2}{2\lambda} \right. \\
\left. + \sum_{\Omega_{\text{in}}^{\text{old}}} \frac{(y - \mu_{\text{in}}^{\text{og}})^2}{2\lambda} - \sum_{\Omega_{\text{out}}^{\text{old}}} \frac{(y - \mu_{\text{out}}^{\text{og}})^2}{2\lambda} \right\}
\]

where the sum is performed over the set of pixels, which are influenced by the change in the segment \( v_{\text{seg}} \). \( \Omega_{\text{in}}^{\text{new}} \), \( \Omega_{\text{out}}^{\text{new}} \), \( \Omega_{\text{in}}^{\text{old}} \), and \( \Omega_{\text{out}}^{\text{old}} \) separate this set into the pixels which belong to the inside and outside of \( v_{\text{new}} \) and \( v_{\text{old}} \), respectively.

6 Adaptive local mean model.

For many problems the local mean varies over the object and background, and the assumption of one global mean, \( \mu_{\text{in}} \) and \( \mu_{\text{out}} \), is insufficient. Another problem is that the constant means make the likelihood model very sensitive to changes in the gray level intensity \( y \). On the 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:

\[
\frac{P(y|v_{\text{new}}, \theta)}{P(y|v_{\text{old}}, \theta)} = \exp\left\{ \sum_{\Omega_{\text{in}}^{\text{og}}} \frac{(y - \mu_{\text{in}}^{\text{og}})^2}{2\lambda} - \sum_{\Omega_{\text{out}}^{\text{og}}} \frac{(y - \mu_{\text{out}}^{\text{og}})^2}{2\lambda} \right. \\
\left. + \sum_{\Omega_{\text{in}}^{\text{og}}} \frac{(y - \mu_{\text{in}}^{\text{og}})^2}{2\lambda} - \sum_{\Omega_{\text{out}}^{\text{og}}} \frac{(y - \mu_{\text{out}}^{\text{og}})^2}{2\lambda} \right\}
\]

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

7 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 2).

![Figure 2: Image \( y_0 \) of cross-section of pork carcass (512x768) with initial configuration overlaid.](image)

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

To estimate the model parameters, 14 outlines of cross-sections, \( o_0, ..., o_{13} \), have been manually marked in 14 images. The first step is to apply the 2D shape learning algorithm [3] to subsample and align the shapes creating the corresponding template parameters \( v_0, ..., v_{13} \) (see figure 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 8. The mean shape, \( \bar{v} \), is then created as the average of \( v_0, ..., v_{13} \) (see figure 3). Before the ML estimator (17) is used to estimate the empirical weight parameters, the approach has been tested on simulated data with the mean equal to \( \bar{v} \). In general the results indicate a robust and stable method which is not sensitive to small sample sizes. The Maximum Likelihood criteria seems to have a smooth convex shape, making the actual optimization very robust.

The empirical weight parameters \( \alpha \) are estimated to \( \alpha_0 = 22.96 \) and \( \alpha_1 = 3.57 \). A powerful tool to
verify the prior model and the estimated mean, $\bar{\alpha}$, and weights, $\alpha$, is to generate samples from the prior model (15) (see figure 4). When the simulations are compared to the real shapes in figure 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, $\theta$, is the calculation of $\mu_{\text{in}} = 150.60, \mu_{\text{out}} = 30.69, \lambda = 778.05, \rho_v = 0.18$ and $\rho_s = 0.09$ following section 3.3. Based on the limited variation in scale, rotation and shape, the mean shape $\bar{\alpha}$ is chosen to be the only relevant configuration used in the fast search strategy. Using the estimated model parameters the likelihood interpreted filter $f_i(\theta)$ and its Fourier transform $F_i(\theta)$ is created (see figure 5). This completes the off-line training.

Given a new image $y_0$, 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 $f_i(\theta)$ and $y_0$ (see figure 2, 5 and 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 knob window 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 ~ 9 percent of the image. Whereas the Fourier based approach calculates 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.
images, where the pixel intensities differ allot from the estimated average intensities, \( \mu_{in} \) and \( \mu_{out} \). The performance of the global mean is improved to an error of 1.54(±0.48), when an adaptive estimation of \( \mu_{in}, \mu_{out} \) and \( \lambda \) is introduced after the initialization.

8 Conclusion and Discussion.

We have proposed a scheme for applying the Grenander Model to a new problem, which requires minimal manual interaction, except the need for a training set of the actual object. The scheme has successfully been applied to segmentation of cross-sections of pork carcasses. The test results indicate a 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.

The proposed scheme has reduced the needed for manual selection of parameters significantly, but still the number of vertices and the optimization parameters need to be selected. The selection of the number of vertices is non-trivial, because it is a tradeoff between speed and accuracy. In a statistical setting the number of vertices can be selected straightforward by using a model selection criteria like the Akaike Information Criteria. Note that outline based training set make it very easy to change to number of vertices, because of the 2D shape learning algorithm. The manual selection of optimization parameters is a minor problem, because the parameters are quite general.

References