A General Scheme for Training and Optimization of the Grenander Deformable Template Model

Rune Fisker[a], Nette Schultz[a], Nicolae Duta[b] and Jens Michael Carstensen[a]

[a]Department of Mathematical Modelling, Technical University of Denmark, 2800, Denmark
[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. [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 [1, 10, 15] 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 [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 maual 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 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, $\boldsymbol{\theta}$, 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. [7, 13] and further investigated and developed in [8, 9, 12, 16].

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 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:

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

1063-6919/00 $10.00 © 2000 IEEE

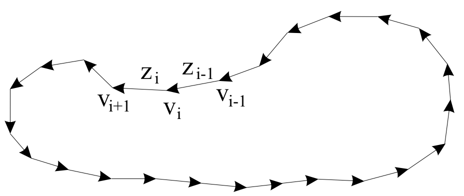

Figure 1: A template with vertices $\boldsymbol{v}$ and edges $\boldsymbol{z}$.

where

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

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, thecorresponding 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:

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

and

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

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

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

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 [7]. The density of a first-order Gaussian-Markov process with mean zero for the n-cyclic parameter vector $\boldsymbol{\kappa}$ is defined by:

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

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

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

$\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$ are such that R_κ^{-1} is positiv edefinite. This parameterization gives an more intuitive interpretation of the parameters controlling R_κ^{-1} than the original parameterization in [7]. The parameters $\boldsymbol{\alpha}$ are easier to interpret in the following form of the density (5):

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

where $c = 2\pi^{n/2}|R_\kappa|^{1/2}$. From (7) it can be seen that α_0 controls the likelihood of how much κ_i differs from zero. For large values of α_0 is very unlikely that κ_i differs very much from zero, because for large values of α_0, $f(\boldsymbol{\kappa}) \approx 0$ even for small values of κ_i. 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 for negative. Large values of α_1 correspond to tight bonding between neighbors.

An equivalent density distribution is derived for the orientationparameter vector ϕ with weight parameters α_0 and α_1. In the original formulation individual weights, $\alpha_{0,i}^\kappa, \alpha_{1,i}^\kappa, \alpha_{0,i}^\phi$ and $\alpha_{1,i}^\phi$, are assigned to each κ_i and ϕ_i, but when the model is actually applied all weights are assumed to be equal $\alpha_{0,i}^\kappa = \alpha_{1,i}^\phi = \alpha_0$ and $\alpha_{1,i}^\kappa = \alpha_{1,i}^\phi = \alpha_1$, see [8, 12, 16]. 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:

$$\boldsymbol{z}_i = \begin{bmatrix} \bar{z}_i^x & -\bar{z}_i^y \\ \bar{z}_i^y & \bar{z}_i^x \end{bmatrix} \begin{bmatrix} \kappa_i \\ \phi_i \end{bmatrix} + \bar{\boldsymbol{z}}_i \tag{8}$$

Rearranging the global vector cycle this way gives:

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

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

$$\Gamma = \begin{bmatrix} \bar{z}_0^x & -\bar{z}_0^y & 0 & 0 & \dots \\ \bar{z}_0^y & \bar{z}_0^x & 0 & 0 & \dots \\ 0 & 0 & \bar{z}_1^x & -\bar{z}_1^y & \dots \\ 0 & 0 & \bar{z}_1^y & \bar{z}_1^x & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad (10)$$

and H is a permutation matrix interchanging rows:

$$H = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad (11)$$

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 (9) for the edges $\boldsymbol{z} \sim N(\bar{\boldsymbol{z}}, R_z)$ where

$$R_z = G R_\omega G^T \quad \text{for} \quad R_\omega = \begin{bmatrix} R_\kappa & 0 \\ 0 & R_\phi \end{bmatrix} \qquad (12)$$

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

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

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

$$R = R_z - R_z L^T (L R_z L^T)^{-1} L R_z \qquad (14)$$

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

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

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

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:

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

where Ω_{in} and Ω_{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 ρ_v and horizontal ρ_h inflation of the smallest rectangle, which contains the actual initial configuration.

3 Model parameter estimation.

To actually apply the Grenander model values need be to 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. 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, $\boldsymbol{a}_{out,0}, ..., \boldsymbol{a}_{out,m-1}$, defining the local areas need to be given.

The central assumption in the estimation of the model parameters of 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 $\boldsymbol{v}_0^t, .., \boldsymbol{v}_{m-1}^t$ 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. A 2D Shape Learning algorithm recently proposed by Duta 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 Procustes distance. This does not insure that the average of the relative size and rotation is one and zero, respectively, which it is suppose to be if the mean template $\bar{v}^t$ 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 $\frac{1}{m}\sum_{i=0}^{m-1}\frac{1}{\gamma_i}$ 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.

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.[14] is one exception. They use a non-general minimax criteria to estimate the regularization parameter. Fisker & Carstensen [5] 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. [12] and Hurn et al. [9] has considered the question of weight parameter estimation in Grenander related models. The work by Kent et al. [12] only considers circulant symmetry objects and the work by Hurn et al. [9] 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 (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:

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

To evaluate $P(\boldsymbol{v}_i|\bar{\boldsymbol{v}}, \boldsymbol{\alpha})$ the inverse of FRF^T need 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. rank$(FRF^T) = 2n-2$. This problem is solved by the use of the following 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 (1). Based on this observation the pseudo inverse $(FRF^T)^-$ is defined by:

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

where Σ_{11} is the first $2n-2$ rows and columns of FRF^T.

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.

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 $\boldsymbol{v}_i^t$.

4 Initialization.

To actually make inference about an object in an image y, estimation of the maximum a posterior (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 [17], the generalized Hough transform [6, 14] and moments [1] 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 [4, 7, 11], where a sparse search is performed in the parameter space Ω_v 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 quit well, if the model should not be caught in a local minimum. For the general setting this make the deterministic search more attractive.

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 (16), can be rearranging into a filtering in the following way (note this is still only for one position of the template):

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

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 the removed need for 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:

1. Create relevant template configurations $\boldsymbol{v}_0, ..., \boldsymbol{v}_{k-1}$.
2. Create likelihood interpreted filters $f_l(\boldsymbol{v}_i,\theta)$ (and optionally their Fourier transform $F_l(\boldsymbol{v}_i,\theta)$).
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 3 if $i < k$.
6. Extract the initial configurations from the calculated values of $P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta})$.

The actual choice and number of initial 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. 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 object in the image are known, the initial configurations are extracted as the corresponding number of local maximas with the highest posterior probability $P(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\theta})_{(r,c)}$. In the case where the number of objects are 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.

5 Optimization

The optimization is performed using a Simulated Annealing scheme incorporating the Metropolis algorithm. The temperature is decreased by an exponential temperature scheme $T_{t+1} = k_T T_t$ 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 $P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{/seg},\theta)$, 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 it neighbors $P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{/seg},\theta) = P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{l-1},\boldsymbol{v}_{l+q},\theta)$. In practice the conditional distribution $P(\boldsymbol{v}_{seg}|\boldsymbol{v}_{k-1},\boldsymbol{v}_{k+q},\theta)$ is obtained from the conditional distributions $f(\boldsymbol{\kappa}_{seg}|\boldsymbol{\kappa}_{k-1},\boldsymbol{\kappa}_{k+q})$ and $f(\boldsymbol{\phi}_{seg}|\boldsymbol{\phi}_{k-1},\boldsymbol{\phi}_{k+q})$, 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

$P(\boldsymbol{v}_{new}|\boldsymbol{y},\theta)/P(\boldsymbol{v}_{old}|\boldsymbol{y},\theta)$. Because most of $\boldsymbol{v}_{new}$ and $\boldsymbol{v}_{old}$ are identical the likelihood ratio reduces to:

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

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 belongs to the inside and outside of $\boldsymbol{v}_{new}$ and $\boldsymbol{v}_{old}$, 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, 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 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:

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

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 in to 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 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).

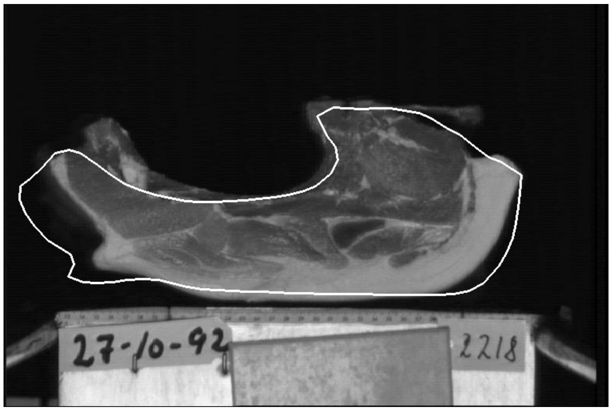

Figure 2: Image $\boldsymbol{y}_0$ of cross-section of pork carcass (512x768) with initial configuration overlaid.

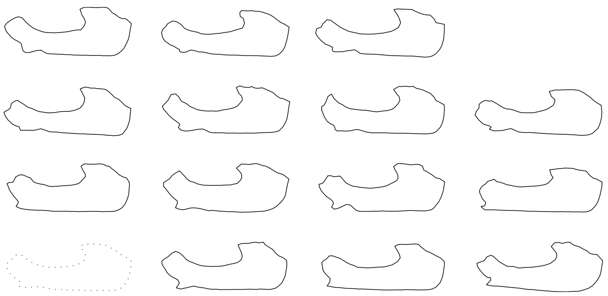

Figure 3: Subsampled and aligned training set (full line) and mean shape (dotted line).

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 [3] to subsample and align the shapes creating the corresponding template parameters $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{13}^t$ (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, $\bar{\boldsymbol{v}}^t$, is then created as the average of $\boldsymbol{v}_0^t, ..., \boldsymbol{v}_{13}^t$ (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 $\bar{\boldsymbol{v}}^t$. 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

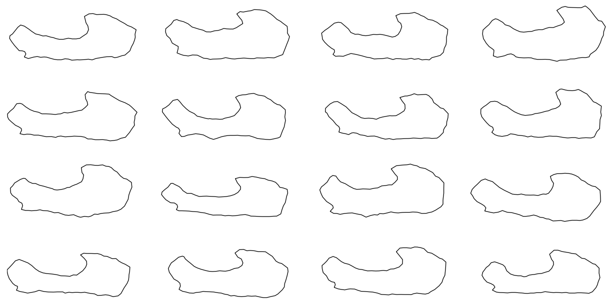

Figure 4: Shape simulations using estimated $\boldsymbol{\alpha}$ and $\bar{\boldsymbol{v}}$.

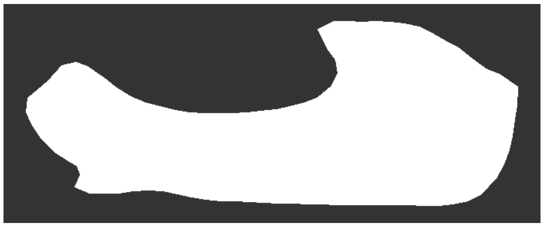

Figure 5: Likelihood interpreted filter $f_l(\boldsymbol{\theta})$ (283x696). The bright and dark color correspond to the pixels inside and outside $\bar{\boldsymbol{v}}^t$ limited by the local area a_{out}.

verify the prior model and the estimated mean, $\bar{\boldsymbol{v}}$, and weights, $\boldsymbol{\alpha}$, 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, $\boldsymbol{\theta}$, 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 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 5). This complete 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 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 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 ~ 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.

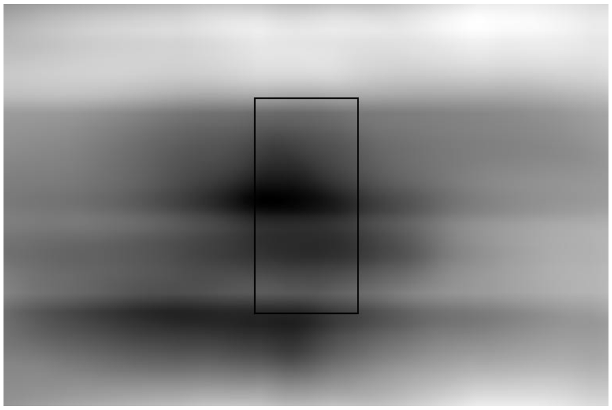

Figure 6: Likelihood energy image. The pixel (r, c) is identical to $U_l(\boldsymbol{y}_0|\bar{\boldsymbol{v}}^t, \boldsymbol{\theta})$ where $\bar{\boldsymbol{v}}^t$ has center in (r, c). The black rectangle corresponds to the ROI.

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 measured from the outline to the template are 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 2 using the adaptive local mean criteria is shown in figure 7. The very high error for the global mean mainly originate from errors in 3

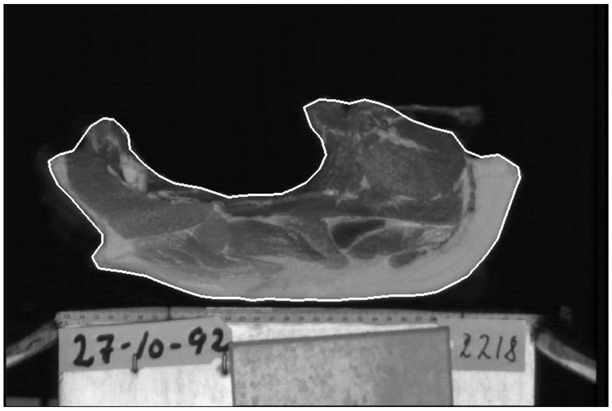

Figure 7: Optimized template using local means.

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 $1.54(\pm 0.48)$, when an adaptive estimation of μ_{in}, μ_{out} and λ 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 straight forward 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

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

[2] 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.

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

[4] 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.

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

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

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

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

[9] 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.

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

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

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

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

[14] K. F. Lai and R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1084–1090, 1995.

[15] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 2(1):91–108, 1996.

[16] N. Schultz and K. Conradsen. 2d vector cycle deformable templates. *Signal Processing*, 71(2):141–153, 1998.

[17] L. H. Staib and S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transaction on Pattern Analysis and Machine intelligence*, 14(11):1061–1075, 1992.