IMAGE MODELLING AND ESTIMATION

A STATISTICAL APPROACH

FINN LINDGREN

Third edition, 2006-12-01, Chapter 7

LUND UNIVERSITY

Centre for Mathematical Sciences
Mathematical Statistics

CENTRUM SCIENTIARUM MATHEMATICARUM

Chapter 7

Free-form shapes

In many applications, it is not possible to construct a single landmark template model which matches all possible objects. In robot vision, it may even be necessary to estimate the shape of objects of types which have never been observed before, e.g. in (very) remote surveying with autonomous vehicles on Mars.

In this chapter, we give a brief introduction to statistical models for free-form shapes, as well as an example of how to estimate such shapes from images.

Throughout the chapter, $\mathbf{x}(t) \in \mathbb{R}^2$, $t \in [0, 1)$, is a closed curve with interior points $\mathcal{X} = \{\mathbf{u};\ \mathbf{u} \text{ inside the curve } \mathbf{x}(t)\}$. A discrete representation of the is given as a sequence of landmarks $\mathbf{x}_k = \mathbf{x}(t_k)$, $k = 1, \ldots, n$, stored in a vectorised landmark matrix $\mathbf{X}$.

7.1 Snakes

The *Snake* is a model designed using ideas from physics (see Kass et al. (1988)). In its usual formulation, the statistical properties are not easily determined. "Internal" and "external" energies are designed, where the internal energies $W_I(\mathbf{x})$ correspond to Bayesian priors, and the external, $W_E(\mathbf{y}, \mathbf{x})$ correspond to data likelihoods:

$$p(\mathbf{x}|\mathbf{y}) \propto \exp\left(-W_I(\mathbf{x}) - W_E(\mathbf{y}, \mathbf{x})\right).$$

A problem with this formulation, in a statistical framework, is that it specifies the posterior distribution for the true shape $\mathbf{x}$ given the measurements $\mathbf{y}$. Bypassing an explicit statement of the data likelihood makes it difficult to improve the model in a systematic way; we have mixed our prior beliefs with our knowledge of the measurement process. Because of this, we will not use this estimation method (often referred to as *active contours*, due to reasons mentioned below). However, we will use the same principles for constructing the internal energy, and thereby define a prior model for $\mathbf{x}$, and then construct explicit conditional data models. The resulting computations are usually the same, but in this way we can interpret the results in the statistical framework.

The internal "Snake" energy function $W(\mathbf{x})$ is given by

$$2W_{\text{snake}}(\mathbf{x}) = \int_0^1 \alpha_1 \left| \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} \right|^2 \mathrm{d}t + \int_0^1 \alpha_2 \left| \frac{\mathrm{d}^2\mathbf{x}(t)}{\mathrm{d}t^2} \right|^2 \mathrm{d}t,$$

where the first integral is the energy of a "rubber band", with unloaded length 0, and the second integral is a stiffness inducing term. The term "active contours" partly refers to the rubber band effect of pulling the shape inwards.

With a view toward common statistical models, we will use a more general form of the snake model. We introduce a *template curve* $\boldsymbol{\mu}(t)$ (with discretisation M) and let

$$2W(\mathbf{x}) = \int_0^1 \left(\alpha_0 |\mathbf{x}(t) - \boldsymbol{\mu}(t)|^2 + \alpha_1 \left| \frac{\mathrm{d}(\mathbf{x}(t) - \boldsymbol{\mu}(t))}{\mathrm{d}t} \right|^2 + \alpha_2 \left| \frac{\mathrm{d}^2(\mathbf{x}(t) - \boldsymbol{\mu}(t))}{\mathrm{d}t^2} \right|^2 \right) \mathrm{d}t,$$

In practical computations, we need to approximate the continuous curve with a discretised version, leading to the expression

$$2W(\mathbf{x}) \approx (\mathbf{X} - \mathbf{M})^\top \mathbf{Q} (\mathbf{X} - \mathbf{M}),$$

where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}' & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}' \end{bmatrix}$$

is a precision matrix, with

$$\mathbf{Q}' = \alpha_0 \mathbf{A}_0 + \alpha_1 \mathbf{W}_1^\top \mathbf{A}_1 \mathbf{W}_1 + \alpha_2 \mathbf{W}_2^\top \mathbf{A}_0 \mathbf{W}_2.$$

Here, the matrices $\mathbf{A}_0$ and $\mathbf{A}_1$ are diagonal matrices of integration weights, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are discrete approximations of the first and second order derivative operators on the discretisation. Thus, this defines a Gaussian Markov random field,

$$\mathbf{X} \in \mathrm{N}(\mathbf{M}, \mathbf{Q}^{-1}),$$

that we can use as a prior model for shape estimation. If $\alpha_0 = 0$, we have an intrinsic field, and if $\mathbf{M} = \mathbf{0}$ we obtain the classical snake model.

7.2 Estimation algorithm example

In this section, we present a simple data model, and an accompanying estimation algorithm.

7.2.1 The data model

The measurement model for the pixels of an image $\mathbf{y}$ is given by the densities

$$p(\tilde{\mathbf{y}}_{\mathbf{u}} = \mathbf{y}_{\mathbf{u}}|\mathbf{X}) = \begin{cases} p_0(\mathbf{y}_{\mathbf{u}}), & \mathbf{u} \notin \mathcal{X}, \\ p_1(\mathbf{y}_{\mathbf{u}}), & \mathbf{u} \in \mathcal{X}, \end{cases}$$

with independence between different pixels.

Example 7.1. The pixels are independent Gaussian random variables in $\mathbb{R}^D$. Pixels outside the shape are $\mathrm{N}(\boldsymbol{\mu}_0, \Sigma_0)$, and pixels inside the shape are $\mathrm{N}(\boldsymbol{\mu}_1, \Sigma_1)$, so that the densities are

$$\begin{aligned} p_0(\mathbf{y}_u) &= \frac{1}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_0}} \exp\left(-\frac{1}{2}(\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_0)^\mathsf{T} \boldsymbol{\Sigma}_0^{-1} (\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_0) \right) \\ p_1(\mathbf{y}_u) &= \frac{1}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_0}} \exp\left(-\frac{1}{2}(\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_1)^\mathsf{T} \boldsymbol{\Sigma}_1^{-1} (\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_1) \right), \end{aligned}$$

with negative logarithms

$$\begin{aligned} -\ln(p_0(\mathbf{y}_u)) &= \frac{D}{2}\ln(2\pi) + \frac{1}{2}\ln(\det \Sigma_0) + \frac{1}{2}(\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_0)^\mathsf{T} \boldsymbol{\Sigma}_0^{-1} (\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_0) \\ -\ln(p_1(\mathbf{y}_u)) &= \frac{D}{2}\ln(2\pi) + \frac{1}{2}\ln(\det \Sigma_1) + \frac{1}{2}(\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_1)^\mathsf{T} \boldsymbol{\Sigma}_1^{-1} (\mathbf{y}_{\mathbf{u}} - \boldsymbol{\mu}_1) \end{aligned}$$

□

7.2.2 Construction of the loss function

The prior shape density for the vectorised landmarks is given by

$$\pi(\mathbf{X}) = \frac{\sqrt{\det \mathbf{Q}}}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}(\mathbf{X} - \mathbf{M})^\mathsf{T} \mathbf{Q} (\mathbf{X} - \mathbf{M}) \right),$$

with negative logarithm

$$-\ln(\pi(\mathbf{X})) = -\frac{1}{2}\det \mathbf{Q} + \frac{n}{2}\ln(2\pi) + \frac{1}{2}(\mathbf{X} - \mathbf{M})^\mathsf{T} \mathbf{Q} (\mathbf{X} - \mathbf{M}).$$

The posterior density for the shape given an image can be obtained through Bayes' formula,

$$\begin{aligned} p(\mathbf{X}|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{X})\pi(\mathbf{X})}{p(\mathbf{y})} \\ &\propto \pi(\mathbf{X}) \prod_{\mathbf{u}} p(\mathbf{y}_{\mathbf{u}}|\mathbf{X}) \\ &= \pi(\mathbf{X}) \prod_{\mathbf{u} \notin \mathcal{X}} p_0(\mathbf{y}_{\mathbf{u}}) \prod_{\mathbf{u} \in \mathcal{X}} p_1(\mathbf{y}_{\mathbf{u}}). \end{aligned}$$

Taking the negative logarithm, we can write

$$\begin{aligned} f(\mathbf{X}) &= \text{constant} - \ln(p(\mathbf{X}|\mathbf{y})) \\ &= \frac{1}{2}(\mathbf{X}-\mathbf{M})^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \sum_{\mathbf{u}\notin\mathcal{X}} z_0(\mathbf{u}) + \sum_{\mathbf{u}\in\mathcal{X}} z_1(\mathbf{u}), \end{aligned}$$

where $z_0(\mathbf{u}) = -\ln(p_0(\mathbf{y}_\mathbf{u}))$ and $z_1(\mathbf{u}) = -\ln(p_1(\mathbf{y}_\mathbf{u}))$. The maximum posterior estimate of $\mathbf{X}$ given $\mathbf{y}$ is the shape that minimises the loss function $f(\mathbf{X})$. We can simplify $f(\mathbf{X})$ by rewriting as follows:

$$\begin{aligned} f(\mathbf{X}) &= \frac{1}{2}(\mathbf{X}-\mathbf{M})^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \sum_{\mathbf{u}} \left(z_0(\mathbf{u})\mathbb{I}(\mathbf{u}\notin\mathcal{X}) + z_1(\mathbf{u})\mathbb{I}(\mathbf{u}\in\mathcal{X})\right) \\ &= \frac{1}{2}(\mathbf{X}-\mathbf{M})^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \sum_{\mathbf{u}} \left(z_0(\mathbf{u})(1-\mathbb{I}(\mathbf{u}\in\mathcal{X})) + z_1(\mathbf{u})\mathbb{I}(\mathbf{u}\in\mathcal{X})\right) \\ &= \frac{1}{2}(\mathbf{X}-\mathbf{M})^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \sum_{\mathbf{u}} \left(z_1(\mathbf{u}) - z_0(\mathbf{u})\right)\mathbb{I}(\mathbf{u}\in\mathcal{X}) + \sum_{\mathbf{u}} z_0(\mathbf{u}), \end{aligned}$$

where the last term is independent of $\mathbf{X}$, and hence need not be included.

7.2.3 Loss function derivatives

First, we approximate the loss function with

$$f(\mathbf{X}) = \frac{1}{2}(\mathbf{X}-\mathbf{M})^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \sum_{\mathbf{u}} z_{01}(\mathbf{u})H_\mathcal{X}(\mathbf{u}),$$

where $z_{01}(\mathbf{u}) = z_1(\mathbf{u}) - z_0(\mathbf{u})$, and H is a smoothed version of the indicator function. If $\mathbf{n}_k$ denotes the outward unit normal vector of the shape at $\mathbf{x}_k$, the local behaviour of $H_\mathcal{X}(\mathbf{u})$ in the vicinity of $\mathbf{x}_k$ is determined by the expression

$$H_\mathcal{X}(\mathbf{u}) \approx \int_{\mathbf{n}_k^\mathsf{T}(\mathbf{u}-\mathbf{x}_k)}^{\infty} h(s)\,\mathrm{d}s,$$

where $h\colon \mathbb{R} \mapsto \mathbb{R}$ is a function centred at 0, such that $\int_{-\infty}^{\infty} h(s)\,\mathrm{d}s = 1$.

The derivative of $H_\mathcal{X}(\mathbf{u})$ with respect to a landmark can thus be approximated by

$$\frac{\partial H_\mathcal{X}(\mathbf{u})}{\partial \mathbf{x}_k} \approx \mathbf{n}_k h(\mathbf{n}_k^\mathsf{T}(\mathbf{u}-\mathbf{x}_k)),$$

and the second derivatives by the matrix

$$\frac{\partial^2 H_\mathcal{X}(\mathbf{u})}{\partial \mathbf{x}_k^2} \approx -\mathbf{n}_k\mathbf{n}_k^\mathsf{T} h'(\mathbf{n}_k^\mathsf{T}(\mathbf{u}-\mathbf{x}_k)),$$

where

$$h'(s) = \frac{\mathrm{d}h(s)}{\mathrm{d}s},$$

and the derivative of $\mathbf{n}_k$ is neglected.

For notational purposes, let $\mathbf{E}_k$ be the derivative of $\mathbf{X}$ with respect to $\mathbf{x}_k$, an $2n \times 2$ matrix with zeros everywhere except for a single 1 in each column, at rows k and $k+n$, respectively.

An approximation of the derivative of the loss function $f(\mathbf{X})$ with respect to landmark $\mathbf{x}_k$ can be obtained by

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{x}_k} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \mathbf{n}_k \sum_{\mathbf{u}} z_{01}(\mathbf{u}) h(\mathbf{n}_k^\mathsf{T}(\mathbf{u}-\mathbf{x}_k)) w_k(\mathbf{u}),$$

where $w_k(\mathbf{u})$ is a weight function with $w_k(\mathbf{x}_k) = 1$, that decreases to zero toward the neighbouring landmarks. The sum can be interpreted as a weighted sum of the z_{01}-function along the shape, near the landmark $\mathbf{x}_k$. With this interpretation, we can modify the approximation to

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{x}_k} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \mathbf{n}_k L_k \sum_{\mathbf{u}} z_{01}(\mathbf{u}) g_0(\mathbf{u}-\mathbf{x}_k),$$

where

$$L_k = \frac{||\mathbf{x}_{k+1}-\mathbf{x}_k|| + ||\mathbf{x}_{k-1}-\mathbf{x}_k||}{2}$$

is the local length measure of the shape. The weight function $g_0(\mathbf{v})$ integrates to 1, so that the resulting sum is a weighted average of $z_{01}(\mathbf{u})$ near $\mathbf{x}_k$. By using the same g_0-function for all derivatives, the sum can be pre-computed by convolving z_{01} and g_0, yielding a function $G_0(\mathbf{u})$, and the loss function derivative can be calculated as

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{x}_k} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \mathbf{n}_k L_k G_0(\mathbf{x}_k) = \mathbf{E}_k^\mathsf{T}\mathbf{Q}(\mathbf{X}-\mathbf{M}) + \mathbf{B}^{[k]}.$$

Proceeding in the same manner for the second derivatives, the second derivatives with respect to a landmark,

$$\frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_{k,d}} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}\mathbf{E}_k - \mathbf{n}_k\mathbf{n}_k^\mathsf{T} \sum_{\mathbf{u}} z_{01}(\mathbf{u}) h'(\mathbf{n}_k^\mathsf{T}(\mathbf{u}-\mathbf{x}_k)) w_k(\mathbf{u}),$$

can be calculated approximately by interpreting the sum as a smooth version of the derivative of $z_{01}(\mathbf{u})$ in the normal direction $\mathbf{n}_k$ near $\mathbf{x}_k$. Using the chain rule for derivatives, we can write

$$\frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_k^2} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}\mathbf{E}_k + \mathbf{n}_k\mathbf{n}_k^\mathsf{T} L_k \left(\mathbf{n}_{k,1} \sum_{\mathbf{u}} z_{01}(\mathbf{u}) g_1(\mathbf{u}-\mathbf{x}_k) + \mathbf{n}_{k,2} \sum_{\mathbf{u}} z_{01}(\mathbf{u}) g_2(\mathbf{u}-\mathbf{x}_k) \right),$$

where g_1 and g_2 are weight functions for derivation along the two axes, near $\mathbf{x}_k$. As before, the sums can be pre-computed, yielding

$$\frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_k^2} \approx \mathbf{E}_k^\mathsf{T}\mathbf{Q}\mathbf{E}_k + \mathbf{n}_k\mathbf{n}_k^\mathsf{T} L_k(\mathbf{n}_{k,1} G_1(\mathbf{x}_k) + \mathbf{n}_{k,2} G_2(\mathbf{x}_k)) = \mathbf{E}_k^\mathsf{T}\mathbf{Q}\mathbf{E}_k + \mathbf{C}^{[k]}.$$

The derivatives of the loss function with respect to the landmarks can be collected into the matrices

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}\mathbf{X}} &= \mathbf{Q}(\mathbf{X}-\mathbf{M}) + \begin{bmatrix}\mathbf{B}_1^{[1]} & \cdots & \mathbf{B}_1^{[p]} & \mathbf{B}_2^{[1]} & \cdots & \mathbf{B}_2^{[p]}\end{bmatrix}^\mathsf{T} \\ \frac{\mathrm{d}^2 f}{\mathrm{d}\mathbf{X}^2} &= \mathbf{Q} + \begin{bmatrix}\operatorname{diag}\left(\left[\mathbf{C}_{1,1}^{[k]}\right]_{k=1,\ldots,p}\right) & \operatorname{diag}\left(\left[\mathbf{C}_{1,2}^{[k]}\right]_{k=1,\ldots,p}\right) \\ \operatorname{diag}\left(\left[\mathbf{C}_{2,1}^{[k]}\right]_{k=1,\ldots,p}\right) & \operatorname{diag}\left(\left[\mathbf{C}_{2,2}^{[k]}\right]_{k=1,\ldots,p}\right)\end{bmatrix}.\end{aligned}$$

7.2.4 Practical optimisation

In the vicinity of the optimal shape (in the maximum Posterior sense), the loss function is approximately quadratic, which makes Newton optimisations feasible, with iteration updates

$$\mathbf{X}^{[i+1]} = \mathbf{X}^{[i]} - \left[\tfrac{\mathrm{d}^2 f}{\mathrm{d}\mathbf{X}^2}(\mathbf{X}^{[i]})\right]^{-1} \left[\tfrac{\mathrm{d}f}{\mathrm{d}\mathbf{X}}(\mathbf{X}^{[i]})\right].$$

In general, however, more robust methods are needed.

Let $\mathbf{D}$ be a matrix of d *allowed directions* for updating $\mathbf{X}$, so that the improved estimate in each iteration takes the form

$$\mathbf{X}^{[i+1]} = \mathbf{X}^{[i]} + \mathbf{D}\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ is a vector of length d. For example, the matrix

$$\mathbf{D} = \begin{bmatrix}\mathbf{1} & \mathbf{0} & \left[\mathbf{x}_{k,1} - \frac{1}{p}\sum_{j=1}^{p}\mathbf{x}_{j,1}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \left[\mathbf{x}_{k,2} - \frac{1}{p}\sum_{j=1}^{p}\mathbf{x}_{j,2}\right]\end{bmatrix}$$

allows translation and scaling along the two coordinate axes. Other $\mathbf{D}$-matrices can be constructed, that allow only large scale changes to the landmark sequence, so that the shape can be kept smooth. Each pair of columns in $\mathbf{D}$ then takes the form

$$\begin{bmatrix}\mathbf{d} & \mathbf{0} \\ \mathbf{0} & \mathbf{d}\end{bmatrix},$$

where $\mathbf{d}$ is a weighting vector. Directions facilitating rotation can also be constructed.

The idea is to first optimise with respect to the large scale shape features, and only permit small scale changes when we are close to the optimal solution.

Assuming that f is quadratic with respect to $\boldsymbol{\theta}$, the optimal choice is given by

$$\boldsymbol{\theta} = -\left(\mathbf{D}^\mathsf{T}\left[\tfrac{\mathrm{d}^2 f}{\mathrm{d}\mathbf{X}^2}(\mathbf{X}^{[i]})\right]\mathbf{D}\right)^{-1}\left(\mathbf{D}^\mathsf{T}\left[\tfrac{\mathrm{d}f}{\mathrm{d}\mathbf{X}}(\mathbf{X}^{[i]})\right]\right).$$

Line-search methods can be used to handle cases where f is not quadratic.

7.2.5 Uncertainty estimation

It would be possible to construct an optimisation that did not rely on the approximative second derivative matrix. However, there is an additional benefit to computing the matrix, due to the fact that the loss function in essence is the negative log-likelihood function. This means that, at the optimum, the inverse of the second derivative matrix,

$$\left(\frac{\mathrm{d}^2 f}{\mathrm{d}\mathbf{X}^2}\right)^{-1},$$

is an estimate of the posterior covariance matrix (compare with the *Fisher information* in classical statistics).

The sparse structure of the approximative $\mathrm{d}^2 f/\mathrm{d}\mathbf{X}^2$ reveals that the posterior distribution is an approximate Gaussian Markov random field.