Active Shape Models - 'Smart Snakes'

T.F.Cootes and C.J.Taylor

Department of Medical Biophysics
University of Manchester
Oxford Road
Manchester M13 9PT
email: bim@wiau.mb.man.ac.uk

Abstract

We describe 'Active Shape Models' which iteratively adapt to refine estimates of the pose, scale and shape of models of image objects. The method uses flexible models derived from sets of training examples. These models, known as Point Distribution Models, represent objects as sets of labelled points. An initial estimate of the location of the model points in an image is improved by attempting to move each point to a better position nearby. Adjustments to the pose variables and shape parameters are calculated. Limits are placed on the shape parameters ensuring that the example can only deform into shapes conforming to global constraints imposed by the training set. An iterative procedure deforms the model example to find the best fit to the image object. Results of applying the method are described. The technique is shown to be a powerful method for refining estimates of object shape and location.

1 Introduction

Flexible models can represent classes of objects whose shape can vary, and can be used to recognise examples of the class in an image. Various authors have described iterative techniques for fitting flexible models to image objects. Kass, Witkin and Terzopoulos [1] described 'Active Contour Models', flexible snakes which can stretch and deform to fit image features to which they are attracted. The iterative energy minimisation technique used is a powerful one, but only simple, local shape constraints are applied. Yuille *et al* [2] describe hand built models consisting of various geometric parts designed to represent image features; they also describe methods for adjusting their models to best fit an image. Unfortunately both the models and the optimisation techniques have to be individually tailored for each application. Staib and Duncan [3] use a Fourier shape model, representing a closed boundary as a sum of trigonometric functions of various frequencies. They too use a form of iterative energy minimisation technique to fit a model to an image. However, using trigonometric functions does not always provide an appropriate basis for capturing shape variability, and is limited to closed boundaries. Lowe [4] describes a technique for fitting projections of three-dimensional parameterised models to two dimensional images by iteratively minimising the distance between lines in the projected model and those in the image.

We have developed a method of building flexible models by representing the objects as sets of labelled points and examining the statistics of their co-ordinates over a number of training shapes - Point Distribution Models (PDMs) [5]. In this

BMVC 1992 doi:10.5244/C.6.28

paper we describe an iterative optimisation scheme for PDMs allowing initial estimates of the pose, scale and shape of an object in an image to be refined. The linear nature of the model leads to simple mathematics allowing rapid execution. Because the models can accurately represent the modes of shape variation of a class of objects they are compact and prevent 'implausible' shapes from occurring. Since PDMs can represent a wide variety of objects the same modelling and refinement framework can be applied in many different applications.

Given an estimate of the position, orientation, scale and shape parameters of an example in an image, adjustments to the parameters can be calculated which give a better fit to the image. Suggested movements are calculated at each model point, giving the displacement required to get to a better location. These movements are transformed to suggested adjustments of the parameters, giving a better overall fit of the model instance to the data. By applying limits to the ranges of the parameters it can be ensured that the shape of the instance remains similar to the original training examples. Enforcing these limits applies global shape constraints, allowing only certain deformations to occur. Because the models attempt to deform to better fit the data, but only in ways which are consistent with the shapes found in the training set we call them 'Active Shape Models' or 'Smart Snakes'.

2 The Point Distribution Model

The Point Distribution Model (PDM) is a way of representing a class of shapes using a flexible model of the position of labelled points placed on examples of the class [5]. The points can represent the boundary or significant internal locations of an object (Figure 1).

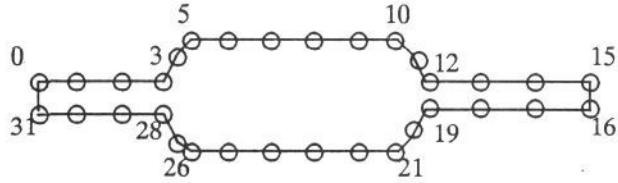


Figure 1 : 32 point model of the boundary of a resistor.

The model consists of the mean positions of these points and the main modes of variation describing how the points tend to move from the mean;

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb} \tag{1}$$

where $\mathbf{x}$ represents the n points of the shape,

$$\mathbf{x} = (x_0,\ y_0,\ x_1,\ y_1,\ \ldots,\ x_k,\ y_k,\ \ldots,\ x_{n-1},\ y_{n-1})^T$$

(x_k, y_k) is the position of point k

$\bar{\mathbf{x}}$ is the mean position of the points

$\mathbf{P} = (\mathbf{p}_1\ \mathbf{p}_2\ \ldots\ \mathbf{p}_t)$ is the matrix of the first t modes of variation, $\mathbf{p}_i$, corresponding to the most significant eigenvectors in a Principal Component Decomposition of the position variables.

$\mathbf{b} = (b_1\ b_2\ \ldots\ b_t)^T$ is a vector of weights for each mode.

The columns of $\mathbf{P}$ are orthogonal so $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ and

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{2}$$

The mean and linearly independent modes of variation are estimated from a set of training examples. The above equations allow us to generate new examples from the class of shapes by varying the parameters (b_i) within suitable limits. The limits are derived by examining the distributions of the parameter values required to generate the training set (typically three standard deviations from the mean). Each parameter varies the global properties of the shape reconstructed.

We can define the shape of a model object, in an object centred co–ordinate frame, by choosing values for $\mathbf{b}$. We can then create an instance, $\mathbf{X}$, of the model in the image frame by defining the position, orientation and scale;

$$\mathbf{X} = M(s,\theta)[\mathbf{x}] + \mathbf{X}_c \tag{3}$$

where $\mathbf{X}_c = (X_c,\ Y_c,\ X_c,\ Y_c,\ \ldots,\ X_c,\ Y_c)^T$

$M(s,\theta)[\ ]$ is a rotation by θ and a scaling by s.

(X_c, Y_c) is the position of the centre of the model in the image frame.

3 Using the PDM as a Local Optimiser - Active Shape Models

Suppose we have a PDM of an object, and we have an estimate of the position, orientation, scale and shape parameters of an example of the object in an image. We would like to improve our estimate, updating the pose and shape parameters to make the model instance fit more accurately to the image evidence. The approach we use is as follows: at each point in the model we calculate a suggested movement required to displace the point to a better position; we calculate the changes to the overall position, orientation and scale of the model which best satisfy the displacements; any residual differences are used to deform the shape of the model object by calculating the required adjustments to the shape parameters. The global shape constraints are enforced by ensuring that the shape parameters remain within appropriate limits.

3.1 Calculating The Suggested Movement of Each Model Point

Given an initial estimate of the positions of a set of model boundary points which we are attempting to fit to the outline of an image object (Figure 2) we need to estimate an adjustment to apply to move each boundary point toward the edge of the image object. There are various approaches that could be taken. In the examples we describe later we use an adjustment along a normal to the model boundary towards the strongest image edge, with a magnitude proportional to the strength of the edge (Figure 3).

A set of adjustments can be calculated, one for each point of the shape (Figure 4). We denote such a set as a vector $d\mathbf{X}$, where

$$d\mathbf{X} = (dX_0,\ dY_0,\ \ldots,\ dX_{n-1},\ dY_{n-1})^T$$

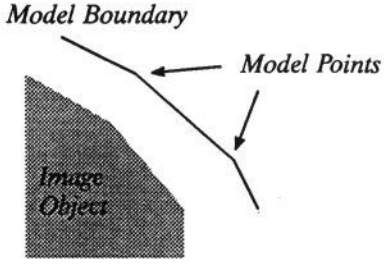


Figure 2 : Part of a model boundary approximating to the edge of an image object.

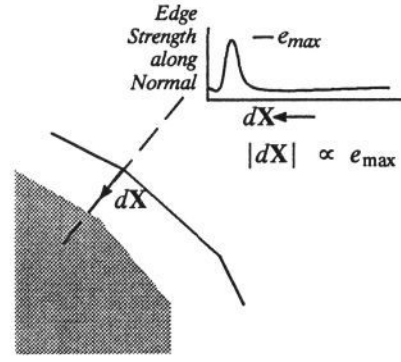


Figure 3 : Suggested movement of point is along normal to boundary, proportional to maximum edge strength on normal.

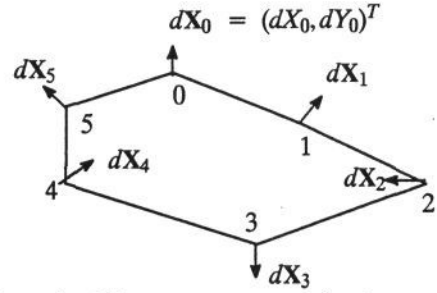


Figure 4 : Adjustments to a set of points

We aim to adjust the pose and shape parameters to move the points from their current locations in the image frame, **X**, to be as close to the suggested new locations (**X** + d**X**) as can be arranged and whilst still satisfying the shape constraints of the model. If the current estimate of the model is centred at (X_c, Y_c) with orientation θ and scale s we would like first to calculate how to update these parameters to better fit the image. One way is to find the translation $(dX_c\ ,\ dY_c)$, rotation $d\theta$ and scaling factor $(1 + ds)$ which best map the current set of points, **X**, onto the set of points given by (**X** + d**X**). Although exact solutions for $dX_c\ ,\ dY_c, d\theta$ and ds are possible [5], we have used the approximation method given in Appendix A, which is quick to calculate and adequate given the iterative nature of the overall scheme.

Having adjusted the pose variables there remain residual adjustments which can only be satisfied by deforming the shape of the model. We wish to calculate the adjustments to the original model points in the local co–ordinate frame **x** required to cause the scaled, rotated and translated points **X** to move by d**X** when combined with the new scale, rotation and translation variables.

The initial position of the points in the image frame is given by

$$\mathbf{X} = M(s, \theta)[\mathbf{x}] + \mathbf{X}_c \qquad (3)$$

We wish to calculate a set of residual adjustments d**x** in the local model co–ordinate frame such that

$$M(s(1 + ds), \theta + d\theta)[\mathbf{x} + d\mathbf{x}] \; + \; (\mathbf{X}_c \; + \; d\mathbf{X}_c) \; = \; (\mathbf{X} + d\mathbf{X}) \tag{4}$$

Thus

$$M(s(1 + ds), \theta + d\theta)[\mathbf{x} + d\mathbf{x}] \; = \; (M(s, \theta)[\mathbf{x}] + d\mathbf{X}) \; - \; (\mathbf{X}_c + d\mathbf{X}_c)$$

and since $$M^{-1}(s, \theta)[\;\;] \; = M(s^{-1}, -\theta)[\;\;]$$

we obtain

$$d\mathbf{x} \; = \; M((s(1 + ds))^{-1}, -(\theta + d\theta))[M(s, \theta)[\mathbf{x}] + d\mathbf{X} - d\mathbf{X}_c] - \mathbf{x} \tag{5}$$

Equation 5 gives a way of calculating the suggested movements to the points $\mathbf{x}$ in the local model co-ordinate frame. These movements are not in general consistent with our shape model. In order to apply the shape constraints we transform $d\mathbf{x}$ into model parameter space giving $d\mathbf{b}$, the changes in model parameters required to adjust the model points as closely to $d\mathbf{x}$ as is allowed by the model. Equation 1 gives

$$\mathbf{x} \; = \; \bar{\mathbf{x}} \; + \; \mathbf{P}\mathbf{b} \tag{1}$$

We wish to find $d\mathbf{b}$ such that

$$\mathbf{x} \; + \; d\mathbf{x} \; \approx \; \bar{\mathbf{x}} \; + \; \mathbf{P}(\mathbf{b} \; + \; d\mathbf{b}) \tag{6}$$

Since there are only t $(<2n)$ modes of variation available and $d\mathbf{x}$ can move the points in $2n$ different degrees of freedom, in general we can only achieve an approximation to the deformation required, since we only allow deformation in the most significant modes observed in the training set. This truncating of the modes of variation is equivalent to setting limits of zero on the parameters controlling other modes of variation. Applying such limits and truncation enforces the global shape constraints.

Subtracting (1) from (6) gives

$$d\mathbf{x} \; \approx \; \mathbf{P}(d\mathbf{b})$$

$$d\mathbf{b} \; = \; \mathbf{P}^T d\mathbf{x} \tag{7}$$

It can be shown that Equation 7 is equivalent to using a least squares approximation to calculate the shape parameter adjustments, $d\mathbf{b}$.

3.2 Updating the Pose and Shape Parameters

The equations above allow us to calculate changes to the pose variables, dX_c, dY_c, $d\theta$ and ds, and adjustments to the shape parameters $d\mathbf{b}$ required to improve the match between an object model and image evidence. We have applied these to update the parameters in an iterative scheme as follows;

$$X_c \; \rightarrow \; X_c \; + \; w_t \; dX_c \tag{8}$$

$$Y_c \; \rightarrow \; Y_c \; + \; w_t \; dY_c \tag{9}$$

$$\theta \; \rightarrow \; \theta \; + \; w_\theta \; d\theta \tag{10}$$

$$s \; \rightarrow \; s(1 \; + \; w_s \; ds) \tag{11}$$

$$\mathbf{b} \rightarrow \mathbf{b} + \mathbf{W}_b \, d\mathbf{b} \tag{12}$$

Where w_t, w_s and w_θ are scalar weights, and $\mathbf{W}_b$ is a diagonal matrix of weights for each mode. This can either be the identity, or each weight can be proportional to the standard deviation of the corresponding shape parameter over the training set. The latter allows more rapid movement in modes in which there tends to be larger shape variation.

In order to ensure that the new shape is plausible it is necessary to apply limits to the b–parameters. If the variance about the origin of the i^{th} parameter over the training set is λ_i then a shape can be considered acceptable if the Mahalanobis distance D_m is less than a suitable constant, D_{max} (for instance 3.0) ;

$$D_m = \sum_{i=1}^{t} \left(\frac{b_i^2}{\lambda_i} \right) \leq D_{\max} \tag{13}$$

(The vector **b** lies within a hyper–ellipsoid about the origin.) If updating **b** using (12) leads to an implausible shape, ie (13) is violated, it can be re–scaled to lie on the closest point of the hyper–ellipsoid using

$$b_i \rightarrow b_i \, . \frac{D_{\max}}{D_m} \quad (i = 1..t) \tag{14}$$

Once the parameters have been updated, and limits applied where necessary, a new example can be calculated, and new suggested movements derived for each point. The procedure is repeated until no significant change results.

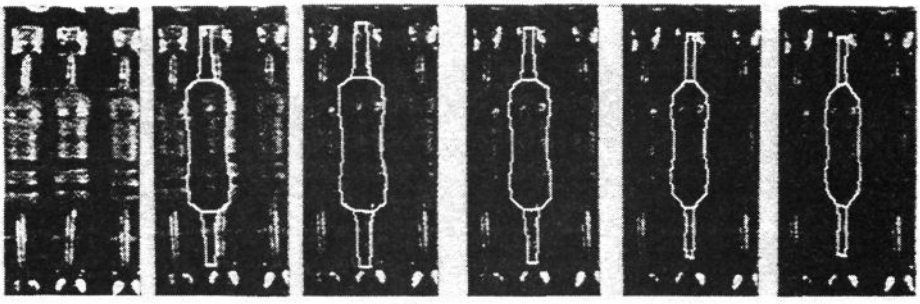
4 Examples Using Active Shape Models

The techniques described above have been used successfully in a number of applications, both industrial and medical [8]. Here we show results obtained using the resistor and hand models described in the companion paper [5].

In both cases initial estimates of the position, orientation and scale are made, and the shape parameters are all initialised at zero ($b_i = 0 \quad (i = 1..t)$). Suggested movements for each model point are calculated by finding the strongest edge (in the correct direction) along the normal to the boundary at the point (See 3.1 and Figure 3). Adjustments to the parameters are calculated and applied, and the process repeated.

4.1 Finding Resistors with an ASM

We have constructed a Point Distribution Model of a resistor representing its boundary using 32 points (Figure 1). Figure 5 shows an image of part of a printed circuit board with the resistor boundary model superimposed as it iterates towards the boundary of a component in the image. We interpolate an additional 32 points, one between each pair of model points around the boundary, and calculate adjustments to each point by finding the strongest edge along profiles 20 pixels long centred at each point. We use a shape model with 5 degrees of freedom. Each iteration takes about 0.025 seconds on a Sun Sparc Workstation.



(a) Original Image *(b) Initial Position* *(c) After 30 iterations* *(d) After 60 iterations* *(e) After 90 iterations* *(f) After 120 iterations*

Figure 5 : Section of Printed Circuit Board with resistor model superimposed, showing its initial position and its location after 30, 60, 90 and 120 iterations.

The ends of the wires are not found correctly since they are not well defined – there is little edge evidence to latch on to. We intend to produce a better model by including the square solder pads. The method is effective in maintaining the global shape constraints of the model and works well given a sufficiently good starting approximation; we discuss methods of obtaining such initial hypotheses elsewhere [6,7,8].

The relatively simple method of calculating the movement of each point, looking for a strong nearby edge, can cause problems when a model is initialised some distance from a component. Highlights and the banding patterns on the resistors can attract the boundary of the model, pulling it away from the true edge. A more sophisticated technique which modelled the banding and possible highlights would be required to overcome this.

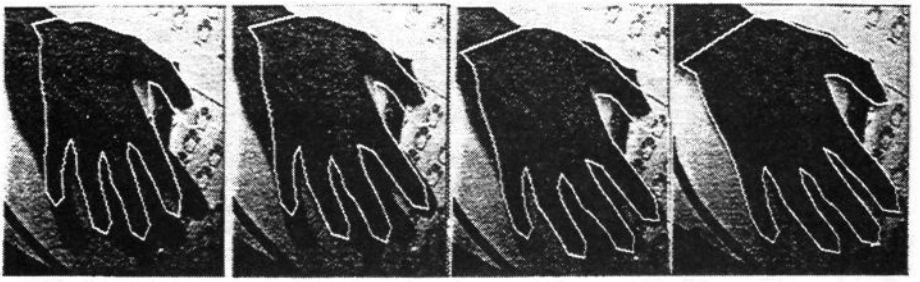
4.2 Finding Hands with an ASM

We have constructed a Point Distribution Model of a hand representing the boundary using 72 points. Figure 6 shows an image of the author's hand and an example of the model iterating towards it. We calculate adjustments to each point by finding the strongest edge on a profile 35 pixels long centred on the point. The shape model has 8 degrees of freedom, and each iteration takes about 0.03 seconds on a Sun Sparc Workstation. The result demonstrates that the method can deal will limited occlusion.

As in the previous example the method works reliably, given a reasonable starting approximation. The example shows that the method is tolerant to quite serious errors in the starting approximation, though this depends on the amount of clutter in the image.

5 Discussion and Conclusions

The iterative approach described above, using image evidence to deform a Point Distribution Model, is effective at locating objects, given an initial estimate of their position, scale and orientation. How good an estimate is required will depend on how cluttered the image is and how well the model describes the object in the image.



(a) Initial Position *(b) 100 iterations* *(c) 200 iterations* *(d) 350 iterations*

Figure 6 : Image of authors hand with hand model superimposed, showing its initial position and its location after 100,200 and 350 iterations.

How suggested adjustments are found for each point is important. Calculating the suggested movement by looking for strong nearby edges is simple and has proved effective in many cases. However, when searching for more complex objects, where the model points do not necessarily lie on strong edges, more sophisticated algorithms are required. Potential maps can be derived, describing how likely each point in an image is to be a particular model point. During a search each model point attempts to move to more likely locations, climbing hills in the potential map. Alternatively a model of the expected grey levels around each model point can be generated from the training examples, and each point moved toward areas which best match its local grey level model. Preliminary experiments using both these techniques have proved promising [9]. The model points do not have to lie only on the boundary of objects, they can represent internal features, and even sub–components of a complex assembly. In the latter case the model describes both the variations in the shapes of the sub–components and the geometric relationships between components. The refinement technique can be applied as easily in this situation as to a model of a single boundary.

By allowing the model to deform, but only in ways seen in the class of examples used as a training set, we have a powerful technique for refinement. The constraints on the shape of the model are applied by the limits on the shape parameters. The $2n–t$ unrepresented modes of variation effectively have limits of zero on their parameters. Rather than fixed limits being used to enforce shape constraints, restoring forces in the parameter space could be applied, pulling the parameters back towards zero against the external 'forces' from the image;

$$\mathbf{b} \rightarrow \mathbf{b} + \mathbf{W}_b d\mathbf{b} - k_b \mathbf{W}_b \mathbf{b} \qquad (0 < k_b < 1) \tag{15}$$

This would give more weight to solutions closer to the mean shape, and require strong evidence for shapes which are considerably deformed. However, this would be likely to lead to compromise solutions between image data and model.

The work we present here can be thought of as a two dimensional application of Lowe's refinement technique [4]. Because of the linear nature of the Point Distribution Model, the mathematics is considerably simpler and can lead to rapid execution.

We have conducted experiments which suggest that the local optimisation method described can be fruitfully used in conjunction with a Genetic Algorithm (GA) search [8]. The GA can be run as a cue generator to produce a number of object hypotheses, which can be refined using the Active Shape Model. Alternatively the ASM can be combined with the GA search, applying one iteration at each generation of the Genetic Algorithm. Both techniques appear very promising.

The method of calculating the parameter changes is straightforward, and new examples of a model can be generated rapidly using linear algebra. As well as the examples of resistors and hand models shown above, the technique has been successfully used in a variety of applications and has great potential for image search in many image analysis domains.

Acknowledgements

This work is funded by SERC under the IEATP Initiative (Project Number 3/2114). The authors would like to thank the other members of the Wolfson Image Analysis Unit for their help and advice, particularly D.H.Cooper, J.Graham, D.Bailes and A.Hill.

Appendix A : Estimating the Pose Parameter Adjustments

Suppose we have a shape defined by the n points in the vector $\mathbf{x}$ relative to the centre of the model, and we wish to find the translation (dX_c, dY_c), rotation about (X_c, Y_c), $d\theta$ and scaling factor $(1+ds)$ which best maps the current set of points, $\mathbf{X}$, onto the set of points given by $(\mathbf{X} + d\mathbf{X})$.

The translation is given by

$$dX_c = \frac{1}{n}\sum_{i=0}^{n-1} dX_i \qquad dY_c = \frac{1}{n}\sum_{i=0}^{n-1} dY_i \tag{16}$$

If we now remove the effects of the translation, letting

$$dX'_i = dX_i - dX_c \qquad dY'_i = dY_i - dY_c \tag{17}$$

$$d\mathbf{X}'_i = (dX'_i,\ dY'_i)^T$$

$$\mathbf{X}' = \mathbf{X} - \mathbf{X}_c \tag{18}$$

then the problem becomes one of finding the rotation $d\theta$ and scaling factor $(1+ds)$ which best maps $\mathbf{X}'$ onto the set of points given by $(\mathbf{X}' + d\mathbf{X}')$.

Consider point i. We wish to move it to point i' (Figure 7)

It is (relatively) easy to show that

$$d\mathbf{X}_{ir} = \frac{X'_i dX'_i + Y'_i dY'_i}{(X'^2_i + Y'^2_i)} \begin{pmatrix} X'_i \\ Y'_i \end{pmatrix} \tag{19}$$

$$d\mathbf{X}_{ia} = d\mathbf{X}'_i - d\mathbf{X}_{ir} \tag{20}$$

$$ds_i = \frac{|d\mathbf{X}_{ir}|}{\sqrt{X'^2_i + Y'^2_i}} = \frac{X'_i dX'_i + Y'_i dY'_i}{(X'^2_i + Y'^2_i)} \tag{21}$$

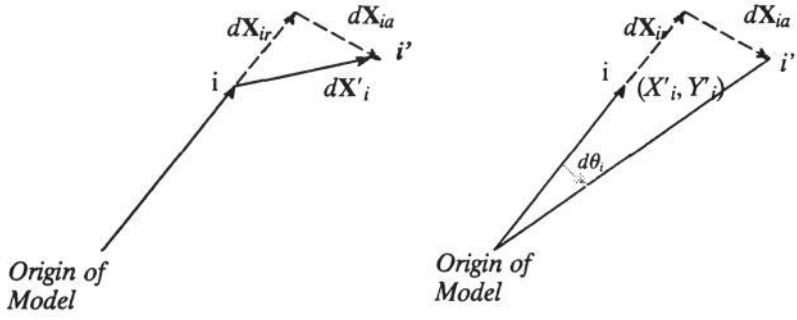


Figure 7 : Estimating the angle and scale changes require to map one point to a new position.

$$d\theta_i = \frac{|dX_{ia}|}{\sqrt{X'^2_i + Y'^2_i}} \tag{22}$$

Then

$$d\theta \approx \frac{1}{n}\sum_{i=0}^{n-1} d\theta_i \tag{23}$$

$$ds \approx \frac{1}{n}\sum_{i=0}^{n-1} ds_i \tag{24}$$

6 References

[1] M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active Contour Models. First International Conference on Computer Vision, pub. IEEE Computer Society Press, 1987; pp 259–268.

[2] A.L. Yuille, D.S. Cohen and P. Hallinan, Feature extraction from faces using deformable templates. Proc. Comp. Vision Patt. Rec. 1989; pp104–109.

[3] L.H. Staib and J.S Duncan, Parametrically Deformable Contour Models. IEEE Computer Society conference on Computer Vision and Pattern Recognition, San Diego, 1989.

[4] D.G. Lowe, Fitting Parameterized Three Dimensional Models to Images. IEEE PAMI 1991; 5, pp441–450.

[5] T.F. Cootes, C.J. Taylor, D.H. Cooper and J. Graham, Training Models of Shape from Sets of Examples. This Volume.

[6] A. Hill and C.J. Taylor, Model-Based Interpretation using Genetic Algorithms. Proc. British Machine Vision Conference, Glasgow, 1991, pub. Springer Verlag, pp266–274.

[7] A. Hill, C.J. Taylor and T.F. Cootes, Object Recognition by Flexible Template Matching using Genetic Algorithms. Proc. European Conference on Computer Vision, Genoa, Italy, 1992.

[8] A. Hill, T.F. Cootes and C.J. Taylor, A Generic System for Image Interpretation Using Flexible Templates. This Volume.

[9] A. Lanitis, Modelling Faces. Internal Progress Report, Wolfson Image Analysis Unit, Manchenster University, 1992.