# Combining Elastic and Statistical Models of Appearance Variation

T.F. Cootes and C.J. Taylor

Department of Imaging Science and Biomedical Engineering,
University of Manchester,
Manchester M13 9PT, U.K.
t.cootes@man.ac.uk
http://www.isbe.man.ac.uk

**Abstract.** We propose a model of appearance and a matching method which combines 'global' models (in which a few parameters control global appearance) with local elastic or optical-flow-based methods, in which deformation is described by many local parameters together with some regularisation constraints. We use an Active Appearance Model (AAM) as the global model, which can match a statistical model of appearance to a new image rapidly. However, the amount of variation allowed is constrained by the modes of the model, which may be too restrictive (for instance when insufficient training examples are available, or the number of modes is deliberately truncated for efficiency or memory conservation). To compensate for this, after global AAM convergence, we allow further local model deformation, driven by local AAMs around each model node. This is analogous to optical flow or 'demon' methods of non-linear image registration. We describe the technique in detail, and demonstrate that allowing this extra freedom can improve the accuracy of object location with only a modest increase in search time. We show the combined method is more accurate than either pure local or pure global model search.

## 1 Introduction

The ability to match a model to an image rapidly and accurately is very important for many problems in computer vision. Here we are concerned with parametrised deformable models, which can synthesise new images of the object(s) represented. For instance, a model of facial appearance should be able to synthesise new face images under a range of simulated viewing conditions. Matching such a model to a new image involves finding the parameters which synthesise something as close as possible to the target image. The model parameters then describe the target object, and can be used for further processing (tracking, measurement, recognition etc).

There have been many attempts to construct such models of appearance. These can be broken down into two broad categories. Those that have a relatively small number of parameters which control *global* appearance properties, and those that have a relatively large number of parameters controlling local appearance properties. The former class includes the Morphable Models of Jones and Poggio [5], the face models of Vetter [13], the shape and appearance models of Cootes *et.al.*[3], the intensity surface models of Nastar *et.al.*[10] and Turk and Pentland's Eigenfaces [12] amongst others. Here changing any one model parameter can change the whole shape or appearance. The second class, that of 'local' models, includes the various non-linear image registration schemes in which the deformation is described by a dense flow field or the positions of a set of grid nodes (these positions are essentially the parameters of the model). Examples of such algorithms are reviewed in [8], and include the work of Christensen [1], Collins *et.al.*[2], Thirion [11] and Lester *et.al.*[7] amongst others.

Global models are not always general enough to represent all new examples accurately. The local models tend not to be specific enough. With no global constraints they can match to illegal examples. Here we propose a model combining both global and local constraints. The global deformation model is a statistical model of appearance as described by Lanitis *et.al.*[6]. This comprises a shape and a texture model trained on a set of landmarked images. A small number of parameters can control both the shape and the texture, allowing approximation of any of the training set and generalisation to unseen examples. Such a model can be matched to a new image rapidly using the 'Active Appearance Model' algorithm [3]. However, without using a large number of model parameters, such a model may not be able to locate accurately a boundary which displays a lot of local variation. To use a large number of parameters can be inefficient, as essentially local shape or texture deformations become represented as global modes of variation.

To overcome this we allow smooth local deformations of the shape around the global model shape. Smoothness is imposed during search by explicitly smoothing any suggested local deformations. Local deformations can be calculated rapidly using a modification of the AAM algorithm in a generalisation of optical flow or Thirion's 'demons' approach [11]. Rather than use local image gradients to drive local neighbourhood matching, we explicitly learn the relationship between local model displacements and the difference image induced. We demonstrate that allowing local deformation can lead to more accurate matching, and that it can be more effective than simply adding more global modes. We describe the algorithm in detail, and show results of experiments testing its performance.

## 2 Background

The algorithm described below is a combination of a deformable statistical model matching algorithm with optical-flow-like image registration techniques. For a comprehensive review of work in these fields there are surveys of image registration methods and deformable models in medical image analysis [9][8]. We give here a brief review of more recent relevant work.

The closest work to ours are the 'Multidimensional Morphable Models' of Jones and Poggio [5]. These are linear models of appearance which can be matched to a new image using a stochastic optimisation method. The model is built from a set of training images using a boot-strapping algorithm. The current model is matched to a new image, optical flow algorithms are used to refine the fit. This gives the correspondences on the new image, allowing it to be added to the model. This boot-strapping step, that of using optical flow style algorithms to improve the match of a model, is similar to the approach we describe in this paper. However, our use of the AAM search algorithm allows much more rapid matching, and we propose that the local search can be an integral part of a multi-resolution search algorithm, rather than just used during model training.

Thirion [11] expresses image registration in a framework of diffusing models, and demonstrates how this can lead to various image matching algorithms. The 'local AAMs' described below are essentially one incarnation of his 'demons'.

Lester *et.al.*[7] describe a non-linear registration algorithm modelling image deformation as fluid flow. They only allow mapping from one image to another, and require a fairly complex prior description of how different parts of the image are allowed to move.

Christensen [1] demonstrates pair-wise image registration using symmetric transformations which are guaranteed to be consistent (ie the mapping from image A to image B is the inverse of that from B to A). He uses a fourier basis for the allowed deformations, and shows how a consistent mean image can be built by repeatedly mapping different images to the current mean. However, the algorithm as described is an image to image, rather than a model to image map.

Collins *et.al.*[2] register two images by building a non-linear deformation field, recursively matching local spherical neighbourhoods. Deformations are only constrained to be locally smooth, imposed by explicit smoothing of the displacement field.

An alternative method of introducing extra flexibility into a linear shape model is to add variation to mimic elastic vibration modes [4], or to add extra artificial covariance to the model to induce such modes [14]. However, this can generate large numbers of global modes which can become too computationally expensive to deal with efficiently. By allowing smooth local deformations, we reduce the computational load considerably.

## 3 Active Appearance Models

### 3.1 Statistical Appearance Models

A statistical appearance model can represent both the shape and texture variability seen in a training set [6, 3]. The training set consists of labelled images, where key landmark points are marked on each example object. For instance, to build a model of the central brain structures in 2D MR images of the brain we need a number of images marked with points at key positions to outline the main features (Figure 1). Similarly a face model requires labelled face images (Figure 2).

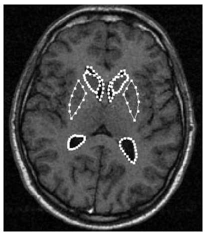


**Fig. 1.** Example of MR brain slice labelled with 123 landmark points around the ventricles, the caudate nucleus and the lentiform nucleus

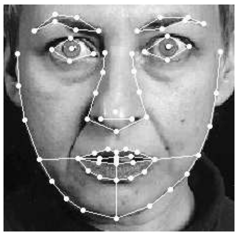


**Fig. 2.** Example of face image labelled with 68 landmark points

Given such a set we can generate a statistical model of shape and texture variation (see [3] for details). Shape can be represented as a vector $\mathbf{x}$ of point positions and the texture (or grey-levels) represented as a vector $\mathbf{g}$. The appearance model has a vector of parameters $\mathbf{c}$ controlling the shape and texture according to

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c} \\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c} \end{aligned} \tag{1}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\bar{\mathbf{g}}$ the mean texture and $\mathbf{Q}_s$,$\mathbf{Q}_g$ are matrices describing the modes of variation derived from the training set.

The model points, $\mathbf{x}$, are in a shape model frame and are mapped into the image frame by applying a euclidean transformation with parameters $\mathbf{t}$, $T_{\mathbf{t}}(\mathbf{x})$. Similarly the texture vector, $\mathbf{g}$, is in a normalised texture frame. The actual image texture is obtained by applying a scaling and offset transformation with parameters, $\mathbf{u}$, $T_{\mathbf{u}}(\mathbf{g})$.

An example image can be synthesised for a given $\mathbf{c}$ by generating a texture image from the vector $\mathbf{g}$ and warping it using the control points described by $\mathbf{x}$ (we currently use a piecewise affine transformation based on a triangulation of the control points). For instance, Figure 3 shows the effects of varying the first two appearance model parameters, $c_1$, $c_2$, of a model trained on a set of face images, labelled as shown in Figure 2. These change both the shape and the texture components of the synthesised image.

## 4 Active Appearance Model Matching

We treat the process of matching an appearance model to an image as an optimisation problem. Given a set of model parameters, $\mathbf{c}$, we can generate a hypothesis

$c_1$ varies by $\pm 2$ s.d.s $\quad\quad$ $c_2$ varies by $\pm 2$ s.d.s

**Fig. 3.** First two modes of an appearance model of a face

for the shape, $\mathbf{x}$, and texture, $\mathbf{g}_m$, of a model instance. To compare this hypothesis with the image, we use the suggested shape to sample the image texture, $\mathbf{g}_s$, and compute the difference, $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$. We seek to minimise the magnitude of $|\delta\mathbf{g}|$.

This is potentially a very difficult optimisation problem, but we exploit the fact that whenever we use a given model with images containing the modelled structure the optimisation problem will be similar. This means that we can learn how to solve the problem off-line. In particular, we observe that the pattern in the difference vector $\delta\mathbf{g}$ will be related to the error in the model parameters.

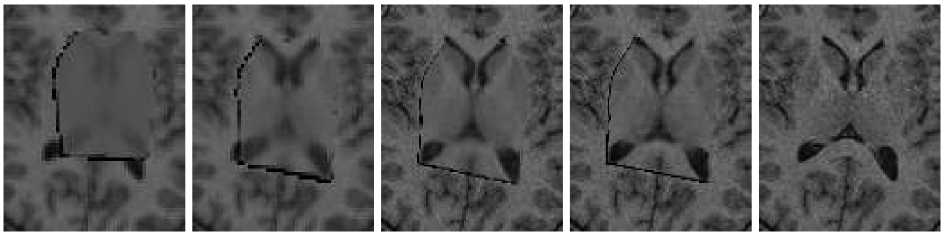
During a training phase, the AAM learns a linear relationship between $\delta\mathbf{g}$ and the parameter perturbation required to drive it to zero, $\delta\mathbf{c} = \mathbf{A}\delta\mathbf{g}$. The matrix $\mathbf{A}$ is obtained by linear regression on random displacements from the known correct model parameters and the induced image residuals (see [3] for details).

During search we simply iteratively compute $\delta\mathbf{g}$ given the current parameters $\mathbf{c}$ and then update the samples using $\mathbf{c} \rightarrow \mathbf{c} - \delta\mathbf{c}$. This is repeated until no improvement is made to the error, $|\delta\mathbf{g}|^2$, and convergence is declared. We use a multi-resolution implementation of search. This is more efficient and can converge to the correct solution from further away than search at a single resolution.

For example, Figure 4 shows an example of an AAM of the central structures of the brain slice converging from a displaced position on a previously unseen image. The model represented about 10000 pixels and $\mathbf{c}$ was a vector of 30 parameters. The search took about 330ms on a 450MHz PC. Figure 5 shows examples of the results of the search, with the found model points superimposed on the target images.

## 5 Allowing Local Deformation

An AAM may not be able to match accurately to a new image if the image displays variation not present in the training set, or if the number of model modes has been truncated too severely. The latter may arise from the need to reduce the storage requirements or search time. In order to match accurately to a new image in such a case, we must allow more deformation than that defined by the model.



Initial  2 its  6 its  16 its  original

**Fig. 4.** Multi-resolution AAM search from a displaced position

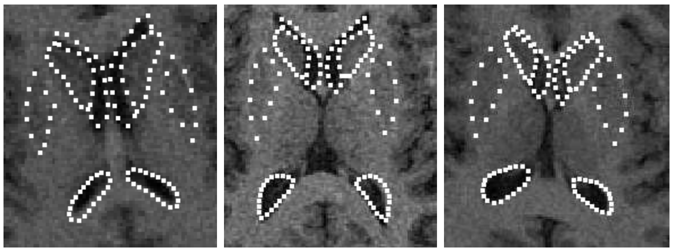


**Fig. 5.** Results of AAM search. Model points superimposed on target image

To allow local deformations, we modify our model of appearance (1) by adding a vector of deformations $\delta\mathbf{x}$ to the shape in the model frame,

$$\begin{aligned}\mathbf{x} &= \bar{\mathbf{x}} + \mathbf{Q}_s\mathbf{c} + \delta\mathbf{x} \\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{Q}_g\mathbf{c}\end{aligned} \tag{2}$$

The allowed local deformations, $\delta\mathbf{x}$, are assumed limited to some range about zero which can be estimated from the training set, and are assumed to represent a smooth deformation from the global model. This assumption of smoothness is necessary to regularise the solution, avoiding overfitting the data.

Valid values of $\delta\mathbf{x}$ can be generated as follows. Let $\mathbf{n}$ be a random vector drawn from a unit spherical normal distribution. Let $\mathbf{M}(\sigma_s)$ be a (sparse) matrix which smooths $\delta\mathbf{x}$ by taking a weighted average over a neighbourhood (see below for its construction).

Then a plausible set of smooth displacements is given by $\delta\mathbf{x} = \alpha\mathbf{M}(\sigma_s)\mathbf{n}$, where $\alpha$ describes the magnitude of the allowed displacements.

For 2D shapes, let the mean shape vector be of the form

$$\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n, \bar{y}_1, \ldots, \bar{y}_n)^T \tag{3}$$

Let $d_{ij}^2 = (\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2$.

We construct the smoothing matrix as

$$\mathbf{M}(\sigma_s) = \begin{pmatrix} \mathbf{M}'(\sigma_s) & \mathbf{0} \\ \mathbf{0} & \mathbf{M}'(\sigma_s) \end{pmatrix},\ M'_{ij}(\sigma_s) = \frac{exp(-d_{ij}^2/2\sigma_s^2)}{\sum_j exp(-d_{ij}^2/2\sigma_s^2)} \tag{4}$$

Since $\mathbf{M}(\sigma_s)$ is symmetric, the covariance of $\delta\mathbf{x}$ is thus $S_\delta = \alpha^2\mathbf{M}(\sigma_s)^2$. We could choose to represent $\delta\mathbf{x}$ using a subset of the eigenvectors of $S_\delta$, $\mathbf{P}_s$, thus $\delta\mathbf{x} = \mathbf{P}_s\mathbf{b}_s$ where $\mathbf{b}_s$ is a set of weights. However, this is a global representation, imposing global constaints. The advantage of the former description is that we can easily update our current estimate of $\delta\mathbf{x}$ to take into account small local changes by smoothing those changes using $\mathbf{M}(\sigma_s)$. This can have computational advantages.

### 5.1 Local AAMs

In order to determine local deformations we use the AAM search algorithm to match image neighbourhoods. For each model point we build a local AAM which will be used to determine the displacement of the point from the position defined by the global model. To construct a neighbourhood we determine all the model pixels within a radius $r$ of the current point in the model frame, and assign them a weight using a gaussian kernel of s.d. $\sigma_w$ centred on the point. For each model point, $i$, let $n_i$ be the number of such pixels, and $n_p$ be the total number of pixels represented by the texture model. Let $\mathbf{W}_i$ be an $n_i$ x $n_p$ sparse matrix (with one non-zero element per row whose value is determined by the gaussian kernel) which can be used to select the weighted subset of pixel values, $\mathbf{g}_i$ from the texture vector $\mathbf{g}$,

$$\mathbf{g}_i = \mathbf{W}_i\mathbf{g} \tag{5}$$

During search, after converging with the conventional AAM (thus finding the optimal pose and model parameters), we compute the residual texture error in the model frame, $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$. For each model point, $i$, we compute the suggested local displacement,

$$\begin{pmatrix} \delta x_i \\ \delta y_i \end{pmatrix} = \mathbf{A}_i\mathbf{W}_i\delta\mathbf{g} \tag{6}$$

where $\mathbf{A}_i$ is a 2 x $n_i$ matrix (for a 2D model) describing the relationship between point displacements and changes in $\mathbf{W}_i\delta\mathbf{g}$, which can be learnt from the training set (see below). This gives a suggested local displacement field, which is smoothed with $\mathbf{M}(\sigma_s)$ before being added to the current estimate of $\delta\mathbf{x}$.

#### Training Local AAMs

We assume that we have trained a global appearance model as described in [3]. For each model point we determine the set of nearby model pixels that will contribute to its movement, and their corresponding weights, $\mathbf{W}_i$.

We then fit the appearance model to each training image, deliberately displace the points, and learn the relationship between the displacements and the induced texture error.

Specifically, for each training image:

1. Fit the global appearance model to the known points
2. For each of a number of small random point displacements, $\delta\mathbf{x}_i$
   (a) Displace the model points by $\delta\mathbf{x}_i$
   (b) Sample the texture vector at the displaced position, $\mathbf{g}_s$
   (c) Compute the texture error, $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$
   (d) For each model point, $i$, compute $\delta\mathbf{g}_i = \mathbf{W}_i \delta\mathbf{g}$

At the end of this process, for each point we have a set of $m$ displacements $\{ (\delta x_i, \delta y_i)_k \}$ $(k = 1..m)$ and a corresponding set of weighted samples, $\{ \delta\mathbf{g}_{i,k} \}$. In the manner of the original AAM, we use multivariate regression to learn the best linear relationship

$$\begin{pmatrix} \delta x_i \\ \delta y_i \end{pmatrix} = \mathbf{A}_i \delta\mathbf{g}_i \tag{7}$$

However, where there is little structure in the texture error vector, it will be difficult to accurately predict the suggested movement. This is equivalent to the difficulty of computing accurate optical flow in smooth regions. We use the same solution as Thirion [11], and weight the contribution by a measure of the amount of local structure. In particular we estimate the relative importance of each point using the total sum of variance of the samples for each point,

$$v_i = \sum_{k=1}^{m} |\delta\mathbf{g}_{i,k}|^2 \tag{8}$$

We then weight each prediction matrix using its relative importance, $\mathbf{A}_i \rightarrow \frac{v_i}{v_{max}}\mathbf{A}_i$ where $v_{max}$ is the largest $v_i$.

The models are trained in a multi-resolution framework (for each point we have a local AAM at each level).

## 6 Matching Algorithm

The full multi-resolution matching algorithm is as follows. We assume that we start with an initial estimate of the pose and the model parameters, $\mathbf{c}$, at the coarsest level. We set $\delta\mathbf{x} = \mathbf{0}$.

For each level, $L = L_{max} \ldots L_{min}$ :

1. Use the standard AAM algorithm to update the pose, $\mathbf{t}$, texture transform, $\mathbf{u}$, and appearance parameters, $\mathbf{c}$, to fit to the image. (for details, see [3]).
2. Use local AAMs to update $\delta\mathbf{x}$ :
   For each of $n_{local}$ iterations:
   (a) Sample image using current shape, $T_{\mathbf{t}}(\mathbf{x})$, giving $\mathbf{g}_{im}$
   (b) Project sample into texture model frame, $\mathbf{g}_s = T_{\mathbf{u}}^{-1}(\mathbf{g}_{im})$
   (c) Compute current model texture, $\mathbf{g}_m$
   (d) Compute current residual $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$
   (e) For each point compute the suggested update $(\delta x'_i, \delta y'_i)^T = -\mathbf{A}_i \mathbf{W}_i \delta\mathbf{g}$

(f) Smooth the suggested updates, $\delta \mathbf{x}_s = \mathbf{M}_s \delta \mathbf{x}'$
(g) Update current displacements: $\delta \mathbf{x} \rightarrow (\delta \mathbf{x} + \delta \mathbf{x}_s)$
(h) Apply limits to displacements

3. If $L \neq L_{min}$ then project pose, $\mathbf{c}$ and $\delta \mathbf{x}$ into level $L-1$.

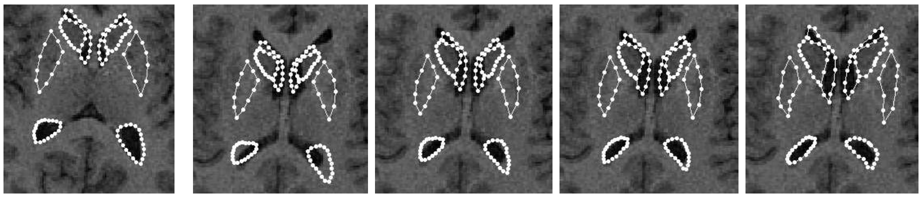
## 7 Experimental Evaluation

We have performed a set of systematic experiments to evaluate the new method and compare its performance to that of the original global AAM search.

### 7.1 Search Using LocalAAMs

In order to test the ability of local AAMs to match a model to an image, we trained a model on a single image (Figure 6a). The global AAM is thus rigid, with no modes of shape or texture variation. The AAM algorithm can still be used to manipulate the pose, $\mathbf{t}$, finding the best euclidean transformation mapping the model to a new image (Figure 6b shows the result of a 3 level multi-resolution search). This does not, however, produce a very accurate fit. We then repeated the search, allowing 2, 5 and 20 local AAM iterations to deform the shape at each level of a three level multi-resolution search. (Figure 6c-e).

This demonstrates that the local AAM algorithm can successfully deform a model to match to an unseen image, giving results similar to those one would expect from optical flow or 'demon' based methods.



| Training Example | Search result (No local it.s) | Search result (2 local it.s) | Search result (5 local it.s) | Search result (20 local it.s) |
|---|---|---|---|---|
| (a) | (b) | (c) | (d) | (e) |

**Fig. 6.** Training a model on a single brain example then searching with differing numbers of local AAM updates. With local deformation the otherwise restricted model can fit well to the new image.

Figure 7 shows a similar experiment, training on one face image then testing on a different one. Because of the shape and texture differences between the original and target images, it is difficult to get an entirely accurate map with a model trained on only one example.

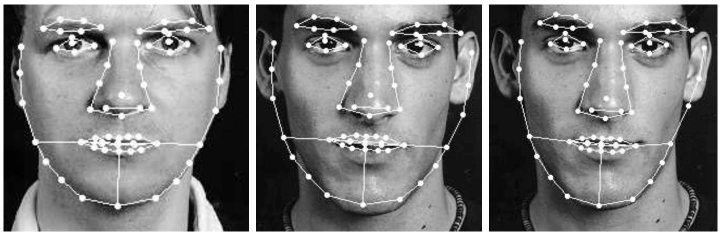



**Fig. 7.** Training a model on a single face example then searching with differing numbers of local AAM updates. Without allowing local deformation, the model cannot fit well to the new image.

### 7.2 Accuracy Improvements using Local AAMs

We wished to quantify the effect on search accuracy of adding local AAMs to the global AAM algorithm. We used two data sets. One contained 68 MR brain scan slices, marked with various sub-cortical structures (Figure 1). The other contained 400 images of 20 different faces, each with different expressions (Figure 2).

For each data set we built models from random subsets and used them to locate the structures of interest on the remaining images. The number of model parameters was chosen so as to explain 95% of the variance observed in the training set. In each case a 3-level multi-resolution search was performed, starting with the mean model parameters, with the model displaced from the true position by $\pm 5$ pixels. On completing the search we measured the RMS texture error between the model and the target image, and the average point position error, measured as the distance from the model point to the boundary on which the point should lie. In the latter case the target boundaries were defined by the original hand-landmarking of the images.

The experiments were performed with a global AAM (allowing no local deformation) and with the modified AAM, allowing 5 local deformation iterations at each resolution.

Figure 8 shows frequency histograms for the position and texture errors for models trained from a single brain example. The results suggest that allowing local deformation improves both position and texture matching. Figure 9 shows equivalent histograms for models trained on ten examples, again showing improvements.

Figure 10 shows graphs of mean error against the number of examples used in the training set for the brain data. In some cases the search diverged - these examples were detected by setting a threshold of 10 pixels on the mean point-to-point error - and they were not included in the calculation of the mean. The graphs demonstrate the use of local AAMs gives a significant improvement in both point location accuracy and the quality of the synthesised texture. (Stan-

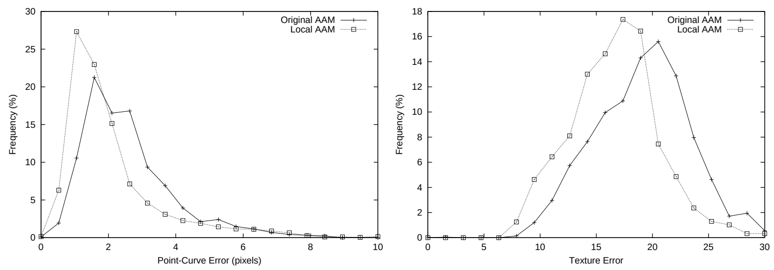



**Fig. 8.** Frequency histograms showing position and texture error distributions for models trained from a single example of a brain

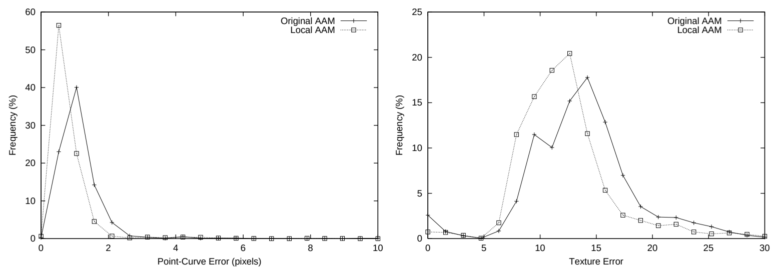



**Fig. 9.** Frequency histograms showing position and texture error distributions for models trained from ten brain images

dard errors on the measurements of the means were too small to be usefully included on the graphs).

Notice that for a model trained on a single example, allowing local deformation significantly improves the match. This is equivalent to various non-linear image registration schemes. However, we can achieve better results using a global AAM (with non local deformation) trained on a small number of examples. The best results are achieved when local and global models are combined together.

Figure 11 shows the equivalent results for the face data. The local AAM again improves the results, but less markedly.

Figure 12 shows typical examples of the mean times per search for differing numbers in the training sets. The experiments were performed on a 550MHz PC. Naturally, allowing local deformation increases the search times, though the matching algorithm is still fast. The extra time is linearly dependent on the number of local iterations used (in this case 5) and the number of pixels in the support region for each model point (in this case a disc of radius 5 pixels).

### 7.3 Varying the model modes

Figure 13 shows the point-to-curve and texture errors as a function of the number of model modes retained when building a global model from subsets of 50

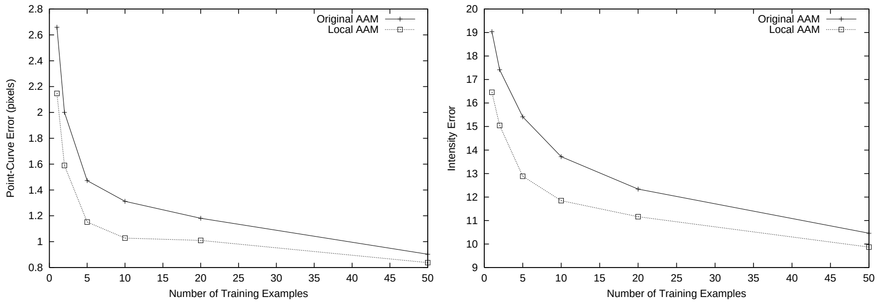



**Fig. 10.** Plots of mean texture error and point position error for models trained on different subset sizes from the brain data set

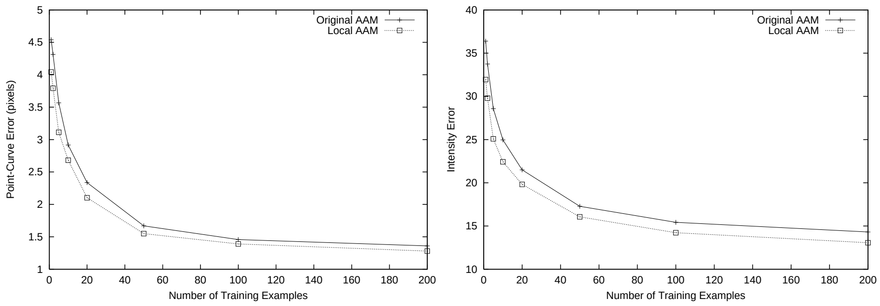


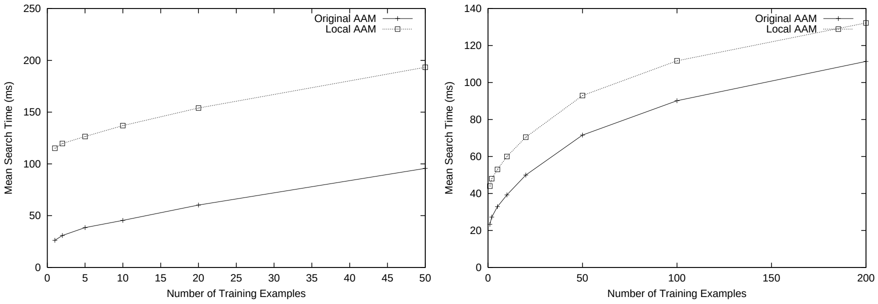

**Fig. 11.** Plots of mean texture error and point position error for models trained on different subset sizes from the face data set

brain images and testing on the remainder. The original AAM performance increases as the number of model modes increases. For the local AAM the allowed movement of the points was estimated from the residual point position variance of the training set not explained by the model modes used. The local AAM gives significantly better results for all choices of numbers of modes. The performance reaches a peak at around 20 modes, after which it begins to deteriorate slightly, probably due to the model becoming over-specific. Clearly graphs of this sort can be used to estimate the optimal numbers of modes to use for a model.

Figure 14 shows the results of equivalent experiments for the face model. Again, these show that the additional local flexibility leads to improvements for all choices of number of modes, and that increasing the number of global modes beyond about 50 can decrease overall performance.

### 7.4 Choice of parameters for local AAMs

There are two significant parameters for the local models, the size of their support region (the radius of the region sampled about each point, $r$) and the standard deviation of the gaussian kernel used to smooth the local displacements, $\sigma_s$. We performed search experiments with various radii of sampled regions. We




Times for brain search Times for face search

**Fig. 12.** Plots of mean time of search for brain and face experiments

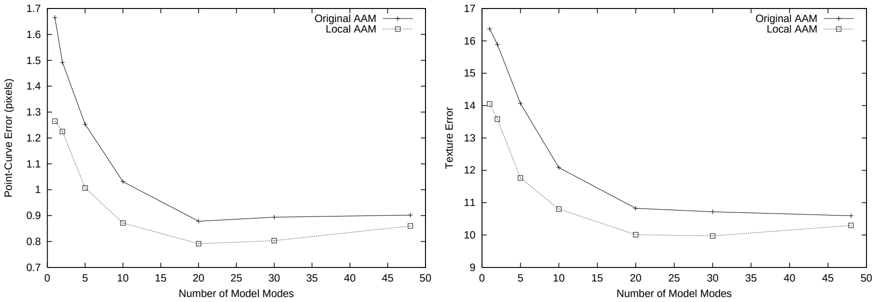



**Fig. 13.** Plot of error against number of brain model modes used

found that for the brain data the results were relatively insensitive to radii in the range $r = 2..10$ pixels.

Figure 15 shows plots of point-to-boundary error for different values of the standard deviation of the local displacement smoothing kernel, $\sigma_s$, for different numbers of training examples. The quality of fit achieved reaches an optimum at about $\sigma_s = 5$ pixels but again is relatively insensitive to variation in this value. More smoothing leads to a degradation, as the model is less able to fit to small changes. With less smoothing, presumably the model becomes too sensitive to noise.

## 8 Discussion and Conclusions

We have demonstrated how the Active Appearance Model can be extended to allow smooth local deformation. Such models can fit more accurately to unseen data than models using purely global model modes or purely local modes. The local AAMs are trained by learning the relationship between point displacements and induced image error (rather than relying on image derivatives), and can match quickly and accurately.

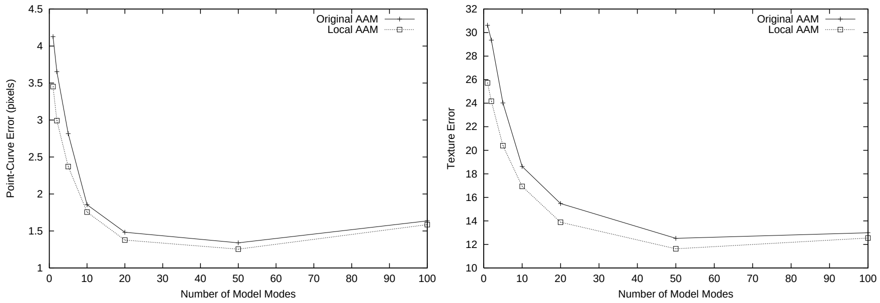



**Fig. 14.** Plot of error against number of face model modes used

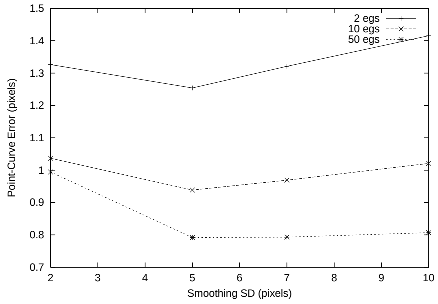



**Fig. 15.** Plot of positional error against SD of local smoothing, $\sigma_s$

The model points used could be a dense grid, rather than the sparce landmarks shown above. We have experimented with this approach. We found it to be slower and no more accurate than using the sparse representation. This is because the landmarks have been deliberately chosen to be those which best describe the shape, and tend to be on structures which can easily be tracked. Were a dense grid used, it would have many points in smooth regions which cannot be accurately tracked - they must be interpolated from nearby structured regions.

We intend to explore using the local AAMs to help automatically build models from sets of images, using a bootstrapping algorithm similar to that described by Jones and Poggio [5].

Allowing local model deformation means that more parameters are required to describe the model state. However, this is justified if a more accurate model fit can be achieved.

The modelling and matching algorithms described above can be considered as a combination of both global appearance model type algorithms and pairwise (local-deformation based) image registration algorithms, and should lead to more robust and accurate solutions in both problem domains.

**Acknowledgements**

Dr Cootes is funded under an EPSRC Advanced Fellowship Grant. The brain images were generated by Dr Hutchinson and colleagues in the Dept. Diagnostic Radiology. They were marked up by Dr Hutchinson, Dr Hill, K. Davies and Prof. A.Jackson(University of Manchester) and Dr G. Cameron (University of Aberdeen). The authors would like to thank the reviewers for their useful comments.

## References

1. G. Christensen. Consistent linear-elastic transformations for image matching. In $16^{th}$ *Conference on Information Processing in Medical Imaging*, pages 224–237, Visegrád, Hungary, June 1999.
2. D. L. Collins, A. Zijdenbos, W. F. C. Baare, and A. C. Evans. Animal+insect: Improved cortical structure segmentation. In $16^{th}$ *Conference on Information Processing in Medical Imaging*, pages 210–223, Visegrád, Hungary, June 1999.
3. T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H.Burkhardt and B. Neumann, editors, $5^{th}$ *European Conference on Computer Vision*, volume 2, pages 484–498. Springer, Berlin, 1998.
4. T. F. Cootes and C. J. Taylor. Combining point distribution models with shape models based on finite-element analysis. *Image and Vision Computing*, 13(5):403–409, 1995.
5. M. J. Jones and T. Poggio. Multidimensional morphable models : A framework for representing and matching object classes. *International Journal of Computer Vision*, 2(29):107–131, 1998.
6. A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.
7. H. Lester, S. A. Arridge, K. M. Jansons, L. Lemieux, J. V. Hajnal, and A. Oatridge. Non-linear registration with the variable viscosity fluid algorithm. In $16^{th}$ *Conference on Information Processing in Medical Imaging*, pages 238–251, Visegrád, Hungary, June 1999.
8. J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
9. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.
10. C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In $4^{th}$ *European Conference on Computer Vision*, volume 1, pages 589–598, Cambridge, UK, 1996.
11. J. P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.
12. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
13. T. Vetter. Learning novel views to a single face image. In $2^{nd}$ *International Conference on Automatic Face and Gesture Recognition 1997*, pages 22–27, Los Alamitos, California, Oct. 1996. IEEE Computer Society Press.
14. Y. Wang and L. H. Staib. Elastic model based non-rigid registration incorporating statistical shape information. In *MICCAI*, pages 1162–1173, 1998.