Active Appearance Models

T.F. Cootes, G.J. Edwards, and C.J. Taylor

Wolfson Image Analysis Unit,
Department of Medical Biophysics,
University of Manchester,
Manchester M13 9PT, U.K.
tcootes@server1.smb.man.ac.uk
http://www.wiau.man.ac.uk

Abstract. We demonstrate a novel method of interpreting images using an Active Appearance Model (AAM). An AAM contains a statistical model of the shape and grey-level appearance of the object of interest which can generalise to almost any valid example. During a training phase we learn the relationship between model parameter displacements and the residual errors induced between a training image and a synthesised model example. To match to an image we measure the current residuals and use the model to predict changes to the current parameters, leading to a better fit. A good overall match is obtained in a few iterations, even from poor starting estimates. We describe the technique in detail and give results of quantitative performance tests. We anticipate that the AAM algorithm will be an important method for locating deformable objects in many applications.

1 Introduction

Model-based approaches to the interpretation of images of variable objects are now attracting considerable interest [6][8][10] [11][14] [16][19][20]. They can achieve robust results by constraining solutions to be valid instances of a model. In addition the ability to 'explain' an image in terms of a set of model parameters provides a natural basis for scene interpretation. In order to realise these benefits, the model of object appearance should be as complete as possible - able to synthesise a very close approximation to any image of the target object.

Although model-based methods have proved successful, few of the existing methods use full, photo-realistic models which are matched directly by minimising the difference between the image under interpretation and one synthesised by the model. Although suitable photo-realistic models exist, (e.g. Edwards *et al* [8] for faces), they typically involve a large number of parameters (50-100) in order to deal with variability in the target objects. Direct optimisation using standard methods over such a high dimensional space is possible but slow [12].

This paper appears in Proc. European Conference on Computer Vision 1998 (H.Burkhardt and B. Neumann Ed.s). Vol. 2, pp. 484-498, Springer, 1998.

In this paper, we show a direct optimisation approach which leads to an algorithm which is rapid, accurate, and robust. In our proposed method, we do not attempt to solve a general optimisation each time we wish to fit the model to a new image. Instead, we exploit the fact the optimisation problem is similar each time - we can learn these similarities off-line. This allows us to find directions of rapid convergence even though the search space has very high dimensionality. This approach is similar to that of Sclaroff and Isidoro [18], but uses a statistical rather than 'physical' model.

In this paper we discuss the idea of image interpretation by synthesis and describe previous related work. In section 2 we explain how we build compact models of object appearance which are capable of generating synthetic examples similar to those in a training set. The method can be used in a wide variety of applications, but as an example we will concentrate on interpreting face images. In section 3 we describe the Active Appearance Model algorithm in detail and in 4 demonstrate its performance.

1.1 Interpretation by Synthesis

In recent years many model-based approaches to the interpretation of images of deformable objects have been described. One motivation is to achieve robust performance by using the model to constrain solutions to be valid examples of the object modelled. A model also provides the basis for a broad range of applications by 'explaining' the appearance of a given image in terms of a compact set of model parameters. These parameters are useful for higher level interpretation of the scene. For instance, when analysing face images they may be used to characterise the identity, pose or expression of a face. In order to interpret a new image, an efficient method of finding the best match between image and model is required.

Various approaches to modelling variability have been described. The most common general approach is to allow a prototype to vary according to some physical model. Bajcsy and Kovacic [1] describe a volume model (of the brain) that also deforms elastically to generate new examples. Christensen *et al* [3] describe a viscous flow model of deformation which they also apply to the brain, but is very computationally expensive. Turk and Pentland [20] use principal component analysis to describe face images in terms of a set of basis functions, or 'eigenfaces'. Though valid modes of variation are learnt from a training set, and are more likely to be more appropriate than a 'physical' model, the eigenface is not robust to shape changes, and does not deal well with variability in pose and expression. However, the model can be matched to an image easily using correlation based methods.

Poggio and co-workers [10] [12] synthesise new views of an object from a set of example views. They fit the model to an unseen view by a stochastic optimisation procedure. This is slow, but can be robust because of the quality of the synthesised images. Cootes *et al* [5] describe a 3D model of the grey-level surface, allowing full synthesis of shape and appearance. However, they do not suggest a plausible search algorithm to match the model to a new image. Nastar *at al* [16] describe a related model of the 3D grey-level surface, combining physical and statistical modes of variation. Though they describe a search algorithm, it requires a very good initialisation. Lades *at al* [13] model shape and some grey level information using Gabor jets. However, they do not impose strong shape constraints and cannot easily synthesise a new instance.

Cootes *et al* [6] model shape and local grey-level appearance, using Active Shape Models (ASMs) to locate flexible objects in new images. Lanitis *at al* [14] use this approach to interpret face images. Having found the shape using an ASM, the face is warped into a normalised frame, in which a model of the intensities of the shape-free face is used to interpret the image. Edwards *at al* [8] extend this work to produce a combined model of shape and grey-level appearance, but again rely on the ASM to locate faces in new images. Our new approach can be seen as a further extension of this idea, using all the information in the combined appearance model to fit to the image.

In developing our new approach we have benefited from insights provided by two earlier papers. Covell [7] demonstrated that the parameters of an eigen-feature model can be used to drive shape model points to the correct place. The AAM described here is an extension of this idea. Black and Yacoob [2] use local, hand crafted models of image flow to track facial features, but do not attempt to model the whole face. The AAM can be thought of as a generalisation of this, in which the image difference patterns corresponding to changes in each model parameter are learnt and used to modify a model estimate.

In a parallel development Sclaroff and Isidoro have demonstrated 'Active Blobs' for tracking [18]. The approach is broadly similar in that they use image differences to drive tracking, learning the relationship between image error and parameter offset in an off-line processing stage. The main difference is that Active Blobs are derived from a single example, whereas Active Appearance Models use a training set of examples. The former use a single example as the original model template, allowing deformations consistent with low energy mesh deformations (derived using a Finite Element method). A simply polynomial model is used to allow changes in intensity across the object. AAMs learn what are valid shape and intensity variations from their training set.

Sclaroff and Isidoro suggest applying a robust kernel to the image differences, an idea we will use in later work. Also, since annotating the training set is the most time consuming part of building an AAM, the Active Blob approach may be useful for 'bootstrapping' from the first example.

2 Modelling Appearance

In this section we outline how our appearance models were generated. The approach follows that described in Edwards *et al* [8] but includes extra normalisation and weighting steps. Some familiarity with the basic approach is required to understand the new Active Appearance Model algorithm.

The models were generated by combining a model of shape variation with a model of the appearance variations in a shape-normalised frame. We require a training set of labelled images, where key landmark points are marked on each example object. For instance, to build a face model we require face images marked with points at key positions to outline the main features (Figure 1).

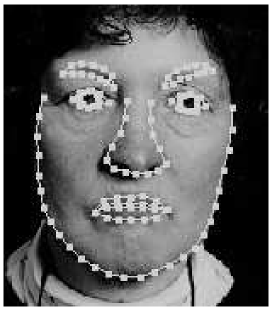

Fig. 1. Example of face image labelled with 122 landmark points

Given such a set we can generate a statistical model of shape variation (see [6] for details). The labelled points on a single object describe the shape of that object. We align all the sets into a common co-ordinate frame and represent each by a vector, $\mathbf{x}$. We then apply a principal component analysis (PCA) to the data. Any example can then be approximated using:

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

where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P}_s$ is a set of orthogonal *modes of variation* and $\mathbf{b}_s$ is a set of shape parameters.

To build a statistical model of the grey-level appearance we warp each example image so that its control points match the mean shape (using a triangulation algorithm). We then sample the grey level information $\mathbf{g}_{im}$ from the *shape-normalised* image over the region covered by the mean shape. To minimise the effect of global lighting variation, we normalise the example samples by applying a scaling, α, and offset, β,

$$\mathbf{g} = (\mathbf{g}_{im} - \beta \mathbf{1})/\alpha \tag{2}$$

The values of α and β are chosen to best match the vector to the normalised mean. Let $\bar{\mathbf{g}}$ be the mean of the normalised data, scaled and offset so that the sum of elements is zero and the variance of elements is unity. The values of α and β required to normalise $\mathbf{g}_{im}$ are then given by

$$\alpha = \mathbf{g}_{im} . \bar{\mathbf{g}} \ \ , \ \ \beta = (\mathbf{g}_{im} . \mathbf{1})/n \tag{3}$$

where n is the number of elements in the vectors.

Of course, obtaining the mean of the normalised data is then a recursive process, as the normalisation is defined in terms of the mean. A stable solution can be found by using one of the examples as the first estimate of the mean, aligning the others to it (using 2 and 3), re-estimating the mean and iterating.

By applying PCA to the normalised data we obtain a linear model:

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \tag{4}$$

where $\bar{\mathbf{g}}$ is the mean normalised grey-level vector, $\mathbf{P}_g$ is a set of orthogonal *modes of variation* and $\mathbf{b}_g$ is a set of grey-level parameters.

The shape and appearance of any example can thus be summarised by the vectors $\mathbf{b}_s$ and $\mathbf{b}_g$. Since there may be correlations between the shape and grey-level variations, we apply a further PCA to the data as follows. For each example we generate the concatenated vector

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \mathbf{P}_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \mathbf{P}_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix} \tag{5}$$

where $\mathbf{W}_s$ is a diagonal matrix of weights for each shape parameter, allowing for the difference in units between the shape and grey models (see below). We apply a PCA on these vectors, giving a further model

$$\mathbf{b} = \mathbf{Q}\mathbf{c} \tag{6}$$

where $\mathbf{Q}$ are the eigenvectors and $\mathbf{c}$ is a vector of *appearance* parameters controlling both the shape and grey-levels of the model. Since the shape and grey-model parameters have zero mean, $\mathbf{c}$ does too.

Note that the linear nature of the model allows us to express the shape and grey-levels directly as functions of $\mathbf{c}$

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{W}_s \mathbf{Q}_s \mathbf{c} \;\; , \;\; \mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{Q}_g \mathbf{c} \tag{7}$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_g \end{pmatrix} \tag{8}$$

An example image can be synthesised for a given $\mathbf{c}$ by generating the shape-free grey-level image from the vector $\mathbf{g}$ and warping it using the control points described by $\mathbf{x}$.

2.1 Choice of Shape Parameter Weights

The elements of $\mathbf{b}_s$ have units of distance, those of $\mathbf{b}_g$ have units of intensity, so they cannot be compared directly. Because $\mathbf{P}_g$ has orthogonal columns, varying $\mathbf{b}_g$ by one unit moves $\mathbf{g}$ by one unit. To make $\mathbf{b}_s$ and $\mathbf{b}_g$ commensurate, we must estimate the effect of varying $\mathbf{b}_s$ on the sample $\mathbf{g}$. To do this we systematically displace each element of $\mathbf{b}_s$ from its optimum value on each training example, and sample the image given the displaced shape. The RMS change in $\mathbf{g}$ per unit change in shape parameter b_s gives the weight w_s to be applied to that parameter in equation (5).

2.2 Example: Facial Appearance Model

We used the method described above to build a model of facial appearance. We used a training set of 400 images of faces, each labelled with 122 points around the main features (Figure 1). From this we generated a shape model with 23 parameters, a shape-free grey model with 114 parameters and a combined appearance model with only 80 parameters required to explain 98% of the observed variation. The model uses about 10,000 pixel values to make up the face patch.

Figures 2 and 3 show the effects of varying the first two shape and grey-level model parameters through ± 3 standard deviations, as determined from the training set. The first parameter corresponds to the largest eigenvalue of the covariance matrix, which gives its variance across the training set. Figure 4 shows the effect of varying the first four appearance model parameters, showing changes in identity, pose and expression.

Fig. 2. First two modes of shape variation (± 3 sd)

Fig. 3. First two modes of grey-level variation (± 3 sd)

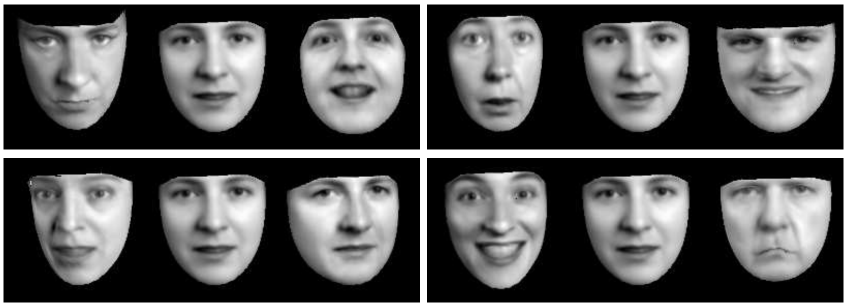

Fig. 4. First four modes of appearance variation (± 3 sd)

2.3 Approximating a New Example

Given a new image, labelled with a set of landmarks, we can generate an approximation with the model. We follow the steps in the previous section to obtain $\mathbf{b}$, combining the shape and grey-level parameters which match the example. Since $\mathbf{Q}$ is orthogonal, the combined appearance model parameters, $\mathbf{c}$ are given by

$$\mathbf{c} = \mathbf{Q}^T\mathbf{b} \tag{9}$$

The full reconstruction is then given by applying equations (7), inverting the grey-level normalisation, applying the appropriate pose to the points and projecting the grey-level vector into the image.

For example, Figure 5 shows a previously unseen image alongside the model reconstruction of the face patch (overlaid on the original image).

Fig. 5. Example of combined model representation (right) of a previously unseen face image (left)

3 Active Appearance Model Search

We now address the central problem: We have an image to be interpreted, a full appearance model as described above and a reasonable starting approximation. We propose a scheme for adjusting the model parameters efficiently, so that a synthetic example is generated, which matches the new image as closely as possible. We first outline the basic idea, before giving details of the algorithm.

3.1 Overview of AAM Search

We wish to treat interpretation as an optimisation problem in which we minimise the difference between a new image and one synthesised by the appearance model. A difference vector $\delta\mathbf{I}$ can be defined:

$$\delta\mathbf{I} = \mathbf{I_i} - \mathbf{I_m} \tag{10}$$

where $\mathbf{I_i}$ is the vector of grey-level values in the image, and $\mathbf{I_m}$, is the vector of grey-level values for the current model parameters.

To locate the best match between model and image, we wish to minimise the magnitude of the difference vector, $\Delta = |\delta\mathbf{I}|^2$, by varying the model parameters, $\mathbf{c}$. Since the appearance models can have many parameters, this appears at first to be a difficult high-dimensional optimisation problem. We note, however, that each attempt to match the model to a new image is actually a similar optimisation problem. We propose to learn something about how to solve this class of problems in advance. By providing a-priori knowledge of how to adjust the model parameters during during image search, we arrive at an efficient run-time algorithm. In particular, the spatial pattern in $\delta\mathbf{I}$, encodes information about how the model parameters should be changed in order to achieve a better fit. In adopting this approach there are two parts to the problem: learning the relationship between $\delta\mathbf{I}$ and the error in the model parameters, $\delta\mathbf{c}$ and using this knowledge in an iterative algorithm for minimising Δ.

3.2 Learning to Correct Model Parameters

The simplest model we could choose for the relationship between $\delta\mathbf{I}$ and the error in the model parameters (and thus the correction which needs to be made) is linear:

$$\delta\mathbf{c} = \mathbf{A}\delta\mathbf{I} \tag{11}$$

This turns out to be a good enough approximation to achieve acceptable results. To find $\mathbf{A}$, we perform multiple multivariate linear regression on a sample of known model displacements, $\delta\mathbf{c}$, and the corresponding difference images, $\delta\mathbf{I}$. We can generate these sets of random displacements by perturbing the 'true' model parameters for the images in which they are known. These can either be the original training images or synthetic images generated with the appearance model. In the latter case we know the parameters exactly, and the images are not corrupted by noise.

As well as perturbations in the model parameters, we also model small displacements in 2D position, scale, and orientation. These four extra parameters are included in the regression; for simplicity of notation, they can be regarded simply as extra elements of the vector $\delta\mathbf{c}$. To retain linearity we represent the pose using (s_x, s_y, t_x, t_y) where $s_x = s\cos(\theta)$, $s_y = s\sin(\theta)$. In order to obtain a well-behaved relationship it is important to choose carefully the frame of reference in which the image difference is calculated. The most suitable frame of reference is the shape-normalised patch described in section 2.

We calculate a difference thus: Let $\mathbf{c}_0$ be the known appearance model parameters for the current image. We displace the parameters by a known amount, $\delta\mathbf{c}$, to obtain new parameters $\mathbf{c} = \delta\mathbf{c} + \mathbf{c}_0$. For these parameters we generate the shape, $\mathbf{x}$, and normalised grey-levels, $\mathbf{g}_m$, using (7). We sample from the image, warped using the points, $\mathbf{x}$, to obtain a normalised sample $\mathbf{g}_s$. The sample error is then $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$.

The training algorithm is then simply to randomly displace the model parameter in each training image, recording $\delta \mathbf{c}$ and $\delta \mathbf{g}$. We then perform multi-variate regression to obtain the relationship

$$\delta \mathbf{c} = \mathbf{A} \delta \mathbf{g} \tag{12}$$

The best range of values of $\delta \mathbf{c}$ to use during training is determined experimentally. Ideally we seek to model a relationship that holds over as large a range errors, $\delta \mathbf{g}$, as possible. However, the real relationship is found to be linear only over a limited range of values. Our experiments on the face model suggest that the optimum perturbation was around 0.5 standard deviations (over the training set) for each model parameter, about 10% in scale and 2 pixels translation.

Results For The Face Model We applied the above algorithm to the face model described in section 2.2. After performing linear regression, we can calculate an R^2 statistic for each parameter perturbation, δc_i to measure how well the displacement is 'predicted' by the error vector $\delta \mathbf{g}$. The average R^2 value for the 80 parameters was 0.82, with a maximum of 0.98 (the 1st parameter) and a minimum of 0.48.

We can visualise the effects of the perturbation as follows. If $\mathbf{a}_i$ is the i^{th} row of the regression matrix $\mathbf{A}$, the predicted change in the i^{th} parameter, δc_i is given by

$$\delta c_i = \mathbf{a}_i . \delta \mathbf{g} \tag{13}$$

and $\mathbf{a}_i$ gives the weight attached to different areas of the sampled patch when estimating the displacement. Figure 6 shows the weights corresponding to changes in the pose parameters, (s_x, s_y, t_x, t_y). Bright areas are positive weights, dark areas negative. As one would expect, the x and y displacement weights are similar to x and y derivative images. Similar results are obtained for weights corresponding to the appearance model parameters

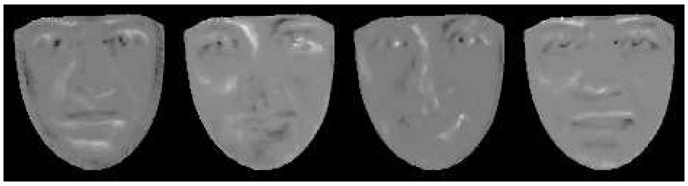

Fig. 6. Weights corresponding to changes in the pose parameters, (s_x, s_y, t_x, t_y)

Perturbing The Face Model To examine the performance of the prediction, we systematically displaced the face model from the true position on a set of 10 test images, and used the model to predict the displacement given the sampled error vector. Figures 7 and 8 show the predicted translations against the actual translations. There is a good linear relationship within about 4 pixels of zero. Although this breaks down with larger displacements, as long as the prediction has the same sign as the actual error, and does not over-predict too far, an iterative updating scheme should converge. In this case up to 20 pixel displacements in x and about 10 in y should be correctable.

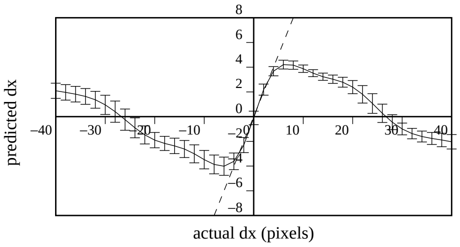

Fig. 7. Predicted dx vs actual dx. Errorbars are 1 standard error

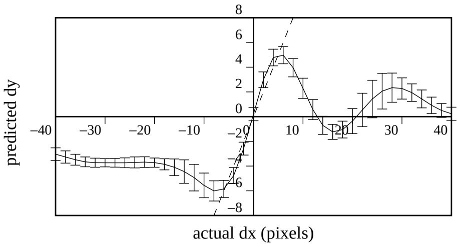

Fig. 8. Predicted dy vs actual dy. Errorbars are 1 standard error

We can, however, extend this range by building a multi-resolution model of object appearance. We generate Gaussian pyramids for each of our training images, and generate an appearance model for each level of the pyramid. Figure 9 shows the predictions of models displaced in x at three resolutions. L0 is the base model, with about 10,000 pixels. L1 has about 2,500 pixels and L2 about 600 pixels.

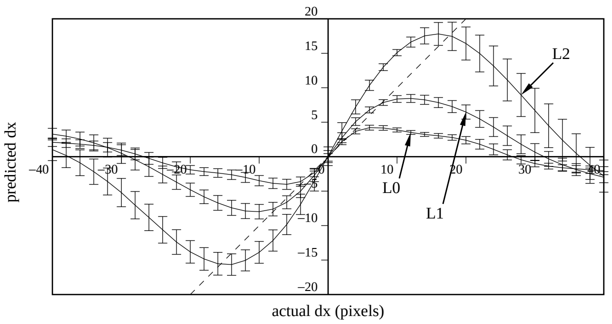

Fig. 9. Predicted dx vs actual dx for 3 levels of a Multi-Resolution model. L0: 10000 pixels, L1: 2500 pixels, L2: 600 pixels. Errorbars are 1 standard error

The linear region of the curve extends over a larger range at the coarser resolutions, but is less accurate than at the finest resolution. Similar results are obtained for variations in other pose parameters and the model parameters.

3.3 Iterative Model Refinement

Given a method for predicting the correction which needs to made in the model parameters we can construct an iterative method for solving our optimisation problem.

Given the current estimate of model parameters, $\mathbf{c}_0$, and the normalised image sample at the current estimate, $\mathbf{g}_s$, one step of the iterative procedure is as follows:

- Evaluate the error vector $\delta \mathbf{g}_0 = \mathbf{g}_s - \mathbf{g}_m$
- Evaluate the current error $E_0 = |\delta \mathbf{g}_0|^2$
- Compute the predicted displacement, $\delta \mathbf{c} = \mathbf{A} \delta \mathbf{g}_0$
- Set $k = 1$
- Let $\mathbf{c}_1 = \mathbf{c}_0 - k \delta \mathbf{c}$
- Sample the image at this new prediction, and calculate a new error vector, $\delta \mathbf{g}_1$
- If $|\delta \mathbf{g}_1|^2 < E_0$ then accept the new estimate, $\mathbf{c}_1$,
- Otherwise try at $k = 1.5$, $k = 0.5$, $k = 0.25$ etc.

This procedure is repeated until no improvement is made to the error, $|\delta \mathbf{g}|^2$, and convergence is declared.

We use a multi-resolution implementation, in which we iterate to convergence at each level before projecting the current solution to the next level of the model. This is more efficient and can converge to the correct solution from further away than search at a single resolution.

Examples of Active Appearance Model Search We used the face AAM to search for faces in previously unseen images. Figure 10 shows the best fit of the model given the image points marked by hand for three faces. Figure 11 shows frames from a AAM search for each face, each starting with the mean model displaced from the true face centre.

Fig. 10. Reconstruction (left) and original (right) given original landmark points

As an example of applying the method to medical images, we built an Appearance Model of part of the knee as seen in a slice through an MR image. The model was trained on 30 examples, each labelled with 42 landmark points.

Initial 2 its 8 its 14 its 20 its converged

Fig. 11. Multi-Resolution search from displaced position

Figure 12 shows the effect of varying the first two appearance model parameters. Figure 13 shows the best fit of the model to a new image, given hand marked landmark points. Figure 14 shows frames from an AAM search from a displaced position.

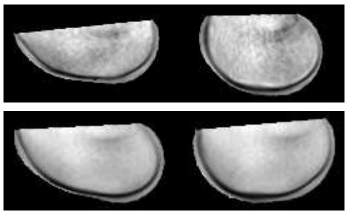

Fig. 12. First two modes of appearance variation of knee model

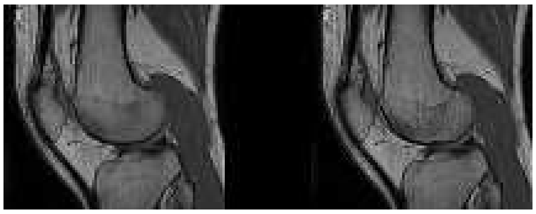

Fig. 13. Best fit of knee model to new image given landmarks

Initial 2 its Converged (11 its)

Fig. 14. Multi-Resolution search for knee

4 Experimental Results

To obtain a quantitative evaluation of the performance of the algorithm we trained a model on 88 hand labelled face images, and tested it on a different set of 100 labelled images. Each face was about 200 pixels wide.

On each test image we systematically displaced the model from the true position by ± 15 pixels in x and y, and changed its scale by $\pm 10\%$. We then ran the multi-resolution search, starting with the mean appearance model. 2700 searches were run in total, each taking an average of 4.1 seconds on a Sun Ultra. Of those 2700, 519 (19%) failed to converge to a satisfactory result (the mean point position error was greater than 7.5 pixels per point). Of those that did converge, the RMS error between the model centre and the target centre was $(0.8, 1.8)$ pixels. The s.d. of the model scale error was 6%. The mean magnitude of the final image error vector in the normalised frame relative to that of the best model fit given the marked points, was 0.88 (sd: 0.1), suggesting that the algorithm is locating a better result than that provided by the marked points. Because it is explicitly minimising the error vector, it will compromise the shape if that leads to an overall improvement of the grey-level fit.

Figure 15 shows the mean intensity error per pixel (for an image using 256 grey-levels) against the number of iterations, averaged over a set of searches at a single resolution. In each case the model was initially displaced by up to 15 pixels. The dotted line gives the mean reconstruction error using the hand marked landmark points, suggesting a good result is obtained by the search.

Figure 16 shows the proportion of 100 multi-resolution searches which converged correctly given starting positions displaced from the true position by up to 50 pixels in x and y. The model displays good results with up to 20 pixels (10% of the face width) displacement.

5 Discussion and Conclusions

We have demonstrated an iterative scheme for fitting an Active Appearance Model to new images. The method makes use of learned correlation between model-displacement and the resulting difference image. Given a reasonable initial starting position, the search converges quickly. Although it is slower than an

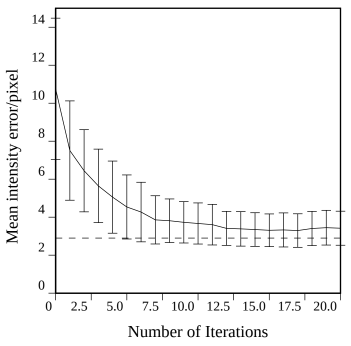

Fig. 15. Mean intensity error as search progresses. Dotted line is the mean error of the best fit to the landmarks.

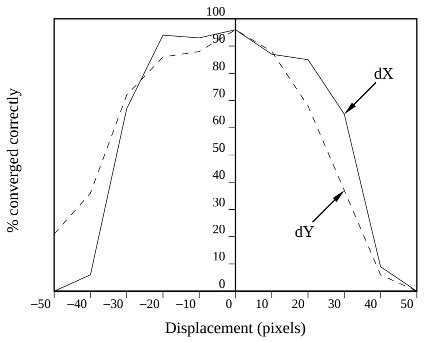

Fig. 16. Proportion of searches which converged from different initial displacements

Active Shape Model [6], since all the image evidence is used, the procedure should be more robust than ASM search alone. We are currently investigating further efficiency improvements, for example, sub-sampling both model and image.

The algorithm can be thought of as a differential optic flow method, in which we learn the patterns of changes associated with varying each parameter. Like differential optic flow, it can only cope with relatively small changes (though the training phase makes it more robust). To deal with larger displacements we are exploring techniques akin to correlation -based optic flow, in which sub-regions of the model are systematically displaced to find the best local correction.

We are attempting to find the parameters $\mathbf{c}$ of some vector valued model $\mathbf{v}(\mathbf{c})$ which minimises $\Delta = |\mathbf{v}_{im} - \mathbf{v}(\mathbf{c})|^2$, where $\mathbf{v}_{im}$ may vary as $\mathbf{c}$ varies. With no other information, this would be difficult, but could be tackled with general purpose algorithms such as Powells, Simplex, Simulated Annealing or Genetic Algorithms [17]. However, by obtaining an estimate of the derivative, $\frac{\partial \mathbf{x}}{\partial \mathbf{c}}$ we can direct the search more effectively. The algorithm described above is related to steepest gradient descent, in which we use our derivative estimate, combined with the current error vector, to determine the next direction to search. It may be possible to modify the algorithm to be more like a conjugate gradient descent method, or to use second order information to use the Levenberg-Marquardt algorithm [17], which could lead to faster convergence.

The nature of the search algorithm makes it suitable for tracking objects in image sequences, where it can be shown to give robust results [9]. In the experiments above we have examined search from relatively large displacements. In practise, a good initial starting point can be found by a variety of methods. We could use an ASM, which by searching along profiles can converge from large displacements. Alternatively we could train a rigid eigen-feature type model [15] [4] which can be used to locate the object using correlation. A few iterations of the AAM would then refine this initial estimate.

We anticipate that the AAM algorithm will be an important method of locating deformable objects in many applications.

References

1. Bajcsy and A. Kovacic. Multiresolution elastic matching. *Computer Graphics and Image Processing*, 46:1–21, 1989.
2. M. J. Black and Y. Yacoob. Recognizing facial expressions under rigid and non-rigid facial motions. In 1^{st} *International Workshop on Automatic Face and Gesture Recognition 1995*, pages 12–17, Zurich, 1995.
3. G. E. Christensen, R. D. Rabbitt, M. I. Miller, S. C. Joshi, U. Grenander, T. A. Coogan, and D. C. V. Essen. *Topological Properties of Smooth Anatomic Maps*, pages 101–112. Kluwer Academic Publishers, 1995.
4. T. Cootes, G. Page, C. Jackson, and C. Taylor. Statistical grey-level models for object location and identification. *Image and Vision Computing*, 14(8):533–540, 1996.
5. T. Cootes and C. Taylor. Modelling object appearance using the grey-level surface. In E. Hancock, editor, 5^{th} *British Machine Vison Conference*, pages 479–488, York, England, September 1994. BMVA Press.
6. 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, January 1995.
7. M. Covell. Eigen-points: Control-point location using principal component analysis. In 2^{nd} *International Conference on Automatic Face and Gesture Recognition 1997*, pages 122–127, Killington, USA, 1996.
8. G. J. Edwards, C. J. Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In 8^{th} *British Machine Vison Conference*, Colchester, UK, 1997.
9. G. J. Edwards, C. J. Taylor, and T. Cootes. Face recognition using the active appearance model. In 5^{th} *European Conference on Computer Vision*, 1998.
10. T. Ezzat and T. Poggio. Facial analysis and synthesis using image-based models. In 2^{nd} *International Conference on Automatic Face and Gesture Recognition 1997*, pages 116–121, Killington, Vermont, 1996.
11. U. Grenander and M. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society B*, 56:249–603, 1993.
12. M. J. Jones and T. Poggio. Multidimensional morphable models. In 6^{th} *International Conference on Computer Vision*, pages 683–688, 1998.
13. M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburt, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.
14. A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.
15. B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. In 5^{th} *International Conference on Computer Vision*, pages 786–793, Cambridge, USA, 1995.
16. 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.

17. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C (2nd Edition)*. Cambridge University Press, 1992.
18. S. Sclaroff and J. Isidoro. Active blobs. In 6^{th} *International Conference on Computer Vision*, pages 1146–53, 1998.
19. L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992.
20. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.