A Comparative Evaluation of Active Appearance Model Algorithms

T.F. Cootes, G. Edwards and C.J. Taylor
Dept. Medical Biophysics,
Manchester University, UK
email: tcootes@server1.smb.man.ac.uk

Abstract

An Active Appearance Model (AAM) allows complex models of shape and appearance to be matched to new images rapidly. An AAM contains a statistical model of the shape and grey-level appearance of an object of interest The associated search algorithm exploits the locally linear relationship between model parameter displacements and the residual errors between model instance and image. This relationship can be learnt during a training phase. To match to an image we measure the current residuals and use the model to predict changes to the current parameters. The algorithm converges in a few iterations. In this paper we describe variations of the basic algorithm aimed at improving the speed and robustness of search. These include subsampling and using image residuals to drive the shape rather than full appearance model. We show examples of search and give the results of experiments comparing the performance of the different algorithms.

1 Introduction

Model based methods are now widely used in image interpretation. By constraining valid solutions a more robust result can be obtained. Recently models have been developed which represent the full appearance of an object, allowing convincing synthetic images to be generated [3][8] [9][11]. With such models image interpretation can be interpretted as an optimisation problem in which we seek the parameters which minimise the difference between a synthetic model image and the target image. Typically the models will have 50 or more parameters. Optimisation in such a high dimensional space using standard methods is possible but slow [9]. However, by exploiting the relationship between parameter displacements and image differences, a fast algorithm can be developed.

An Active Appearance Model (AAM) contains a statistical model of the shape and grey-level appearance of an object of interest, which can be fit rapidly to an example in a new image [3]. The appearance model, given a good enough training set, can generalise to almost any valid example of the class of objects represented, potentially giving a full photo-realistic approximation.

During a training phase a model instance is randomly displaced from the optimum position in a set of training images. The difference between the displaced model instance and the image is recorded, and linear regression is used to estimate the relationship between this residual and the parameter displacement.

During image search we wish to find the parameters which minimise the difference between image and synthesised model instance. An initial estimate of the instance is placed in the image and the current residuals are measured. The relationship is then used to predict the changes to the current parameters which would lead to a better fit. A good overall match is obtained in a few iterations, even from poor starting estimates. The algorithm is capable of fitting a 10000 pixel, 93 parameter model of a face to a new image in a few seconds.

The AAM, though related to the Active Shape Model (ASM) [5], differs from it by matching a full model of grey-level appearance to a target image. The ASM only located the shape of the modelled objects, not the texture, so was not taking full advantage of the information available.

In this paper we describe modifications to the basic AAM algorithm aimed at improving the speed and robustness of search. Since some regions of the model may change little when parameters are varied, we need only sample the image in regions where significant changes are expected. This should reduce the cost of each iteration.

The original formulation manipulates the combined shape and grey-level parameters directly. An alternative approach is to use image residuals to drive the shape parameters, computing the grey-level parameters directly from the image given the current shape. This approach may be useful when there are few shape modes and many grey-level modes.

In the following we describe the algorithm in more detail, show examples of search and give the results of experiments comparing the effects of the modifications.

2 Background

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* [2] describe a viscous flow model of deformation which they also apply to the brain, but is very computationally expensive.

Turk and Pentland [14] 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 [8] [9] 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* [4] 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* [12] 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* [10] 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* [5] model shape and local grey-level appearance, using Active Shape Models (ASMs) to locate flexible objects in new images. Lanitis *at al* [11] 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* [6] 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.

Sclaroff and Isidoro describe 'Active Blobs' for tracking [13]. The approach is similar to that of the AAM, though Active Blobs are derived from a single example, rather than a training set of examples. The example is used as a template, with low energy elastic shape deformations allowed. A simply polynomial model is used to allow changes in intensity across the object. In contrast, AAMs learn what are valid shape and intensity variations from their training set.

An application of the AAM to tracking and face recognition is described by Edwards [7].

3 Active Appearance Models

This section outlines the basic AAM algorithm. A more comprehensive description is given in [3]. The AAM contains two main components. A parameterised model of object appearance and an estimate of the relationship between parameter displacements and induced image residuals.

3.1 Appearance Models

The appearance model can represent both shape and texture changes seen in a training set. The training set consists 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).

Given such a set we can generate a statistical model of shape variation (see [5] for details). The labelled points, $\mathbf{x}$, on a single object describe the shape of that object. 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 shape variation* and $\mathbf{b}_s$ is a set of shape parameters.

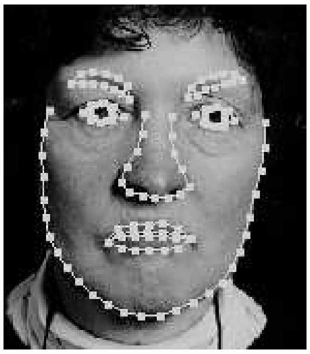

Figure 1: Example of face image labelled with 122 landmark points

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 from the *shape-normalised* image over the region covered by the mean shape. To minimise the effect of global lighting variation, we normalise the resulting samples.

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

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \tag{2}$$

where $\bar{\mathbf{g}}$ is the mean normalised grey-level vector, $\mathbf{P}_g$ is a set of orthogonal *modes of grey 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 concatenate the vectors, apply a further PCA and obtain a model of the form

$$\begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \mathbf{b} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_g \end{pmatrix} \mathbf{c} = \mathbf{Q}\mathbf{c} \tag{3}$$

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, $\mathbf{Q}$ is a set of orthogonal modes 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 $\mathbf{c}$

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{W}_s \mathbf{Q}_s \mathbf{c} \quad , \quad \mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{Q}_g \mathbf{c} \tag{4}$$

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 $\mathbf{x}$.

For instance, Figure 2 shows the effects of varying the first two parameters, c_1, c_2, of an appearance model trained on a set of face images.

c_1 varies by ± 2 s.d.s c_2 varies by ± 2 s.d.s

Figure 2: First two modes of appearance model of a face

3.2 Active Appearance Model Search

We treat interpretation as an optimisation problem in which we minimise the difference between a new image and one synthesised by the appearance model.

Given a set of model parameters, $\mathbf{c}$, we can generate a hypothesis 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 this, $|\delta\mathbf{g}|$.

During a training phase, the AAM learns a linear relationship between $\delta\mathbf{g}$ and the parameter perturbation required to correct this, $\delta\mathbf{c}$,

$$\delta\mathbf{c} = \mathbf{A}\delta\mathbf{g} \tag{5}$$

The matrix $\mathbf{A}$ is obtained by linear regression on random displacements from the true training set positions and the induced image residuals (See [3] for details).

We can use (5) in an iterative search algorithm. Given the current estimate of model parameters, $\mathbf{c}_0$, and the normalised image sample at the current estimate, $\mathbf{g}_s$, each iteration proceeds 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 $k = 0.5$, $k = 0.25$ etc.

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, 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.

For instance, Figure 3 shows examples of an AAM of a face converging from a displaced position on a previously unseen image. This takes about one second (on a Sun Ultra-SPARC) to fit a 93 parameter model representing about 10000 pixels.

Figure 3: Multi-resolution AAM search from a displaced position

3.3 Sub-sampling During Search

In the original formulation, during search we sample all the points in the model to obtain $\mathbf{g}_s$, with which we predict the change to the model parameters. There may be 10000 or more such pixels, but fewer than 100 parameters. There is thus considerable redundancy, and it may be possible to obtain good results by sampling at only a sub-set of the modelled pixels. This could significantly reduce the computational cost of the algorithm.

The change in the i^{th} parameter, δc_i, is given by

$$\delta c_i = \mathbf{A}_i \delta \mathbf{g} \tag{6}$$

Where $\mathbf{A}_i$ is the i^{th} row of $\mathbf{A}$.

The elements of $\mathbf{A}_i$ indicate the significance of the corresponding pixel in the calculation of the change in the parameter. To choose the most useful subset for a given parameter, we simply sort the elements by absolute value and select the largest. However, the pixels which best predict changes to one parameter may not be useful for any other parameter.

To select a useful subset for all parameters we compute the best $u\%$ of elements for each parameter, then generate the union of such sets. If u is small enough, the union will be less than all the elements.

Given such a subset, we perform a new multi-variate regression, to compute the relationship, $\mathbf{A}'$ between the changes in the subset of samples, $\delta \mathbf{g}'$, and the changes in parameters

$$\delta \mathbf{c} = \mathbf{A}' \delta \mathbf{g}' \tag{7}$$

Search can proceed as described above, but using only a subset of all the pixels.

3.4 Search Using Shape Parameters

The original formulation manipulates the parameters, $\mathbf{c}$. An alternative approach is to use image residuals to drive the shape parameters, $\mathbf{b}_s$, computing the grey-level parameters,

$\mathbf{b}_g$, and thus $\mathbf{c}$, directly from the image given the current shape. This approach may be useful when there are few shape modes and many grey-level modes.

The update equation in this case has the form

$$\delta\mathbf{b}_s = \mathbf{B}\delta\mathbf{g} \tag{8}$$

where in this case $\delta\mathbf{g}$ is given by the difference between the current image sample $\mathbf{g}_s$ and the best fit of the grey-level model to it, $\mathbf{g}_m$,

$$\begin{aligned} \delta\mathbf{g} &= \mathbf{g}_s - \mathbf{g}_m \\ &= \mathbf{g}_s - (\bar{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g) \end{aligned} \tag{9}$$

where $\mathbf{b}_g = \mathbf{P}_g^T(\mathbf{g}_s - \bar{\mathbf{g}})$.

During a training phase we use regression to learn the relationship, $\mathbf{B}$, between $\delta\mathbf{b}_s$ and $\delta\mathbf{g}$ (as given in (9)). Since any $\delta\mathbf{g}$ is orthogonal to the columns of $\mathbf{P}_g$, the update equation simplifies to

$$\begin{aligned} \delta\mathbf{b}_s &= \mathbf{B}(\mathbf{g}_s - \bar{\mathbf{g}}) \\ &= \mathbf{B}\mathbf{g}_s - \mathbf{b}_{offset} \end{aligned} \tag{10}$$

Thus one approach to fitting a model to an image is simply to keep track of the pose and shape parameters, $\mathbf{b}_s$. The grey-level parameters can be computed directly from the sample at the current shape. The constraints of the combined appearance model can be applied by computing $\mathbf{c}$ using (3), applying constraints then recomputing the shape parameters. As in the original formulation, the magnitude of the residual $|\delta\mathbf{g}|$ can be used to test for convergence.

In cases where there are significantly fewer modes of shape variation than combined appearance modes, this approach may be faster. However, since it is only indirectly driving the parameters controlling the full appearance, $\mathbf{c}$, it may not perform as well as the original formulation.

Note that we could test for convergence by monitoring changes in the shape parameters, or simply apply a fixed number of iterations at each resolution. In this case we do not need to use the grey-level model at all during search. We would just do a single match to the grey-levels sampled from the final shape. This may give a significantly faster algorithm.

4 Results of Experiments

To compare the variations on the algorithm described above, an appearance model was trained on a set of 300 labelled faces. This set contains several images of each of 40 people, with a variety of different expressions. Each image was hand annotated with 122 landmark points on the key features. From this data was built a shape model with 36 parameters, a grey-level model of 10000 pixels with 223 parameters and a combined appearance model with 93 parameters.

Three versions of the AAM were trained for these models. One with the original formulation, a second using a sub-set of 25% of the pixels to drive the parameters $\mathbf{c}$, and a third trained to drive the shape parameters, $\mathbf{b}_s$, alone.

A test set of 100 unseen new images (of the same set of people but with different expressions) was used to compare the performance of the algorithms. On each image the optimal pose was found from hand annotated landmarks. The model was displaced by (+15, 0 ,-15) pixels in x and y, the remaining parameters were set to zero and a multi-resolution search performed (9 tests per image, 900 in all).

Two search regimes were used. In the first a maximum of 5 iterations were allowed at each resolution level. Each iteration tested the model at $\mathbf{c} \rightarrow \mathbf{c} - k\delta\mathbf{c}$ for $k = 1.0, 0.5^1, \ldots, 0.5^4$, accepting the first that gave an improved result or declaring convergence if none did.

The second regime forced the update $\mathbf{c} \rightarrow \mathbf{c} - \delta\mathbf{c}$ without testing whether it was better or not, applying 5 steps at each resolution level.

The quality of fit was recorded in two ways;

- The RMS grey-level error per pixel in the normalised frame, $\sqrt{|\delta\mathbf{v}|^2/n_{pixels}}$
- The mean distance error per model point

For example, the result of the first search shown in Figure 3 above gives an RMS grey error of 0.46 per pixel and a mean distance error of 3.7 pixels.

Some searches will fail to converge to near the correct result. This is detected by a threshold on the mean distance error per model point. Those that have a mean error of > 7.5 pixels were considered to have failed to converge.

Table 1 summarises the results. The final errors recorded were averaged over those searches which converged successfully.. The top row corresponds to the original formulation of the AAM. It was the slowest, but on average gave the fewest failures and the smallest grey-level error. Forcing the iterations decreased the quality of the results, but was about 25% faster.

Sub-sampling considerably speeded up the search (taking only 30% of the time for full sampling) but was much less likely to converge correctly, and gave a poorer overall result.

Driving the shape parameters during search was faster still, but again lead to more failures than the original AAM. However, it did lead to more accurate location of the target points when the search converged correctly. This was at the expense of increasing the error in the grey-level match.

The best fit of the Appearance Model to the images given the labels gave a mean RMS grey error of 0.37 per pixel over the test set, suggesting the AAM was getting close to the best possible result most of the time.

Driven Params	Sub-sample	Iterations		Failure Rate	Final Errors		Mean Time (ms)
		Max.	Forced		Point ± 0.05	Grey ± 0.005	
$\mathbf{c}$	100%	5	1	4.1%	4.2	0.45	3270
$\mathbf{c}$	100%	5	5	4.6%	4.4	0.46	2490
$\mathbf{c}$	25%	5	1	13.9%	4.6	0.60	920
$\mathbf{c}$	25%	5	5	22.9%	4.8	0.63	630
$\mathbf{b}_s$	100%	5	1	11.4%	4.0	0.85	560
$\mathbf{b}_s$	100%	5	5	11.9%	4.1	0.86	490

Table 1: Comparison between AAM algorithms given displaced centres (See Text)

Table 2 shows the results of a similar experiment in which the models were started from the best estimate of the correct pose, but with other model parameters initialised to zero. This shows a much reduced failure rate, but confirms the conclusions drawn from the first experiment. The search could fail even given the correct initial pose because some of the images contain quite exaggerated expressions and head movements, a long way from the mean. These were difficult to match to, even under the best conditions.

Driven Params	Sub-sample	Iterations		Failure Rate	Final Errors	
		Max.	Forced		Point ± 0.1	Grey ± 0.01
$\mathbf{c}$	100%	5	1	3%	4.2	0.46
$\mathbf{c}$	100%	5	5	4%	4.4	0.47
$\mathbf{c}$	25%	5	1	10%	4.6	0.60
$\mathbf{c}$	25%	5	5	10%	4.6	0.60
$\mathbf{b}_s$	100%	5	1	6%	4.0	0.84
$\mathbf{b}_s$	100%	5	5	6%	4.1	0.87

Table 2: Comparison between AAM algorithms, given correct initial pose. (See Text)

5 Discussion and Conclusions

We have described several modifications that can be made to the Active Appearance Model algorithm. Sub-sampling and driving the shape parameters during search both lead to faster convergence, but were more prone to failure. The shape based method was able to locate the points slightly more accurately than the original formulation. Testing for improvement and convergence at each iteration slowed the search down, but lead to better final results.

It may be possible to use combinations of these approaches to achieve good results quickly, for instance using the shape based search in the early stages, then polishing with the original AAM. Further work will include developing strategies for reducing the numbers of convergence failures and extending the models to use colour or multispectral images.

Though only demonstrated for face models, the algorithm has wide applicability, for instance in matching models of structures in MR images [3]. The AAM algorithms, being able to match 10000 pixel, 100 parameter models to new images in a few seconds or less, are powerful new tools for image interpretation.

References

[1] Bajcsy and A. Kovacic. Multiresolution elastic matching. *Computer Graphics and Image Processing*, 46:1–21, 1989.

[2] 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.

[3] T. 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, 1998.

[4] 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.

[5] 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.

[6] 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*, pages 130–139, Colchester, UK, 1997.

[7] G. J. Edwards, C. J. Taylor, and T. Cootes. Face recognition using the active appearance model. In H.Burkhardt and B. Neumann, editors, 5^{th} *European Conference on Computer Vision*, volume 2, pages 581–695. Springer, 1998.

[8] 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.

[9] M. J. Jones and T. Poggio. Multidimensional morphable models. In 6^{th} *International Conference on Computer Vision*, pages 683–688, 1998.

[10] 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.

[11] 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.

[12] 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.

[13] S. Sclaroff and J. Isidoro. Active blobs. In 6^{th} *International Conference on Computer Vision*, pages 1146–53, 1998.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.