Advances in Active Appearance Models

G. J. Edwards, T. F. Cootes and C. J. Taylor
Division of Imaging Science and Biomedical Engineering
University of Manchester
Manchester
M14 9PT
England

Abstract
This paper presents advances in the construction and use of Active Appearance Models (AAMs) for image interpretation. AAMs are photo-realistic generative models of object appearance that can be used to rapidly locate deformable objects in images. We extend the AAM method to include coloured texture and present an enhanced search algorithm with the ability to locate partially occluded objects. Previously, AAMs have been limited by the need for good manual initialization. In this paper, we describe a hierarchical search algorithm that overcomes this drawback. The extended AAM method provides a complete, unified scheme for model based image interpretation. We demonstrate the application of the scheme to the task of locating faces in images.

Photo-realistic, generative models attempt to represent all object information, including both shape and texture and can be used to generate synthetic reconstructions of objects. Edwards et al [4] described the construction of 2D Appearance Models of faces, which included both shape and grey-level texture. Similar models of faces have been described by Jones and Poggio [7] and Vetter [13]. Vetter and Blanz [14] describe a 3D colour model of faces, which is used to interpret 2D images by projection. In principal, each of these approaches could be used to model other types of variable objects - Cootes et al [1] have described Appearance Models of MRI scans of knee joints.

1 Introduction
Model-based approaches have proved successful in many areas of image interpretation, particularly where the objects of interest are significantly variable in shape, texture, or both. Robust interpretation in the presence of noisy or missing data is achieved by constraining the interpretation to plausible solutions. Explaining image data in terms of model parameters provides a natural basis for further interpretation. The field of Face Recognition has seen extensive use of model-based methods [6] [10] [12] [8], due to the need to deal with the large amount of variability present in images of faces.

The usefulness of a model depends on its specificity - it should only be able to represent plausible instances of the object class. In order to be truly specific, a model must represent all the possible data in images of the modelled objects. For example, models of shape such as the Active Shape Model described by Cootes et al [3] can only generate plausible object shapes, but it is possible to find plausible shapes which enclose invalid regions of texture for the object class of interest.

Using such models for image interpretation requires a method of fitting them to new image data. Jones and Poggio and Vetter and Blanz fit their appearance models using forms of stochastic gradient descent. This is a difficult method to apply to a model containing up to 100 parameters, and image search is very slow (around 10 minutes in Jones and Poggio’s approach). An additional drawback is the need for very good initialisation of the model to ensure convergence. A faster, more robust method known as the Active Appearance Model (AAM) was introduced by Edwards et al [5] and described further by Cootes et al [1]. Their approach uses a prior training stage in which the relationship between model parameter displacements and induced residual errors is learnt. This provides a method of matching high-dimensional models to image data in a fraction of a second. Although the method is much faster than direct optimisation techniques, it also requires fairly good initialisation in the image if the search is to converge. A further shortcoming of the method is the lack of robustness to any type of occlusion. In this paper we present an extended colour AAM which addresses both these issues.
2 Modelling Colour Appearance

In this section we describe the formulation of an Appearance Model which encodes both shape and
coloured texture information. The use of colour aims
to provide further specificity than a single channel
grey-scale model.

2.1 Formulation

A Colour Appearance Model is based on the statistical analysis of both shape and texture information
for a set of training images. The key requirement is to
establish correspondence between pixels in each training
image. In order to do this, we follow the approach
of Edwards et al [4] and place by hand key landmarks on boundary features and other features which can
be consistently found in each of the training images.
Given these key points, a piece-wise affine triangle-

based warping algorithm is used to deform each image to an average shape. The warp is computed and
applied to each colour channel separately. In Figure 3
we show the effect of warping some typical images of
faces to the average face shape.

Each training image can be represented by two vec-
tors - a shape vector and a texture vector. The shape
vector, \( \mathbf{x} \), is given by:

\[
\mathbf{x} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^T
\]

where \((x_i, y_i)\) is the position of the \(i^{th}\) landmark point.
The texture vector, \( \mathbf{v} \), contains the RGB values of each pixel in the shape-normalised image and is given by:

\[
\mathbf{v} = (v_{r1}, v_{r2}, \ldots, v_{rm},
v_{g1}, v_{g2}, \ldots, v_{gm},
v_{b1}, v_{b2}, \ldots, v_{bm})^T
\]

where \(v_{ri}, v_{gi}\) and \(v_{bi}\) are the intensities of the red,
green and blue channels of the \(i^{th}\) pixel in the shape-
free patch.

Both the shape vectors and texture vectors for all
the training samples are aligned to an average shape and
texture. The shape vectors \( \mathbf{x}_i \) are aligned using a
form of Procrustes analysis [3] to minimise variation in
scale, rotation and translation. To minimise the effect
of global lighting variation we align each texture vector
\( \mathbf{v}_i \), by applying a scale, \( \alpha \) and shift, \( \beta \) according to:

\[
\mathbf{v}'_i = (\mathbf{v}_i - \beta \mathbf{1})/\alpha
\]

In a recursive process, we align each example to the
normalised mean of the training data, \( \overline{\mathbf{v}} \) by computing
\( \alpha \) and \( \beta \):

\[
\alpha = \mathbf{v}_i, \overline{\mathbf{v}}
\]

\[
\beta = (\mathbf{v}_i, \mathbf{1})/(3m)
\]

The remaining steps in generating the Appearance
Model are identical to those described by Cootes et al
[1], noting only that the texture vector, \( \mathbf{v} \), contains
colour rather than intensity information. Two stages of Principal Component Analysis yield a model of the form:

\[
\mathbf{x} = \overline{\mathbf{x}} + \mathbf{M}_s \mathbf{c}
\]

\[
\mathbf{v} = \overline{\mathbf{v}} + \mathbf{M}_t \mathbf{c}
\]

where \( \overline{\mathbf{x}} \) and \( \overline{\mathbf{v}} \) are the mean values of the shape and
texture vectors over the training set. The matrices \( \mathbf{M}_s \) and \( \mathbf{M}_t \) describe the relationship between shape and	
texture, and a vector of model parameters, \( \mathbf{c} \).

In this formulation, any object of the model class
can be generated by choosing a value of \( \mathbf{c} \). The
dimensionality of \( \mathbf{c} \) is usually much lower than the
original dimensionality of the data. Figure 4 shows
the effect of varying the first two elements of \( \mathbf{c} \) for a
50 dimensional model of faces.

3 A Robust AAM algorithm

The Active Appearance Model algorithm allows a model of the form given above to be rapidly fitted to
image data. Full details are given by Edwards et al
[5] - here we give a brief overview.

3.1 The AAM algorithm

The iterative scheme uses the difference between a
reconstructed image generated by the model and the
underlying target image, to drive the model parame-
ters towards better values. In a prior learning stage,
known displacements, \( \delta \mathbf{c} \), are applied to known model
instances and the resulting difference between model
and image, \( \delta \mathbf{v} \), is measured. Multivariate linear regres-
sion is applied to a large set of such training dis-
placements and an approximate linear relationship is established:

\[
\delta \mathbf{c} = \mathbf{A} \delta \mathbf{v}
\]

In image search, the current difference between
model and image, \( \delta \mathbf{v} \), is used to predict an adjust-
ment, \(-\delta \mathbf{c}\), to the model parameters which improves
model fit. For simplicity of notation, the vector \( \delta \mathbf{c} \) is
assumed to include displacements in scale, rotation, and translation.

The algorithm is effective in cases where the initial placement of the model is reasonably close to the solution. In the example of faces, convergence is achieved if the initial placement is no more than one-third of a face width from the solution. Convergence is achieved when the measured difference, $\delta v$, tends towards zero. Since the model can only represent complete, plausible faces, $\delta v$, can never become zero in the presence of occlusion and search fails in these cases.

### 3.2 Estimating difference limits

We assume that occlusion will produce a much larger difference between pixels than in the case of a complete object. By observing the magnitude of the elements of $\delta v$ for each training displacement, we can estimate a suitable threshold vector, $v_t$. Elements of the vector $\delta v$ whose magnitude is greater than the corresponding element of $v_t$ are ignored at each search step. A reasonable estimate of $v_t$ is the maximum magnitude of each element of $\delta v$ observed during training. However, since some of the training displacements will result in the displaced model instance overlapping the background, the estimate of $v_t$ will be too high, particularly for those sample points near the edge of the model. To overcome this we ignore displaced model pixels not overlapping a pixel from the real object (for training examples we know the location of the real object pixels exactly). Such pixels are also ignored in the linear regression used to calculate $A$. This has the added advantage of preventing the AAM from overfitting the background in image search - the background in unseen images may be quite different to that in the training set.

We applied the above method to the face model shown in Figure 4. Figure 1 illustrates the learnt difference threshold, $v_t$. Brighter pixels correspond to areas where the model tolerates larger image differences during search, before regarding the difference as due to occlusion. Notice that areas of significant facial structure are more inclined to show large differences during search.

Figure 5 shows an example of image search using the simple AAM algorithm and using the extended robust algorithm. Notice how the simple algorithm locks onto the section of the face that is not occluded.

### 4 Searching Entire Images

The major limitation of AAM search is the need for good initialisation. Here we propose a method for searching an entire image using an Active Appearance Model.

Given a model of object appearance and an image, the aim is to find regions in the image that could be plausible instances of the model. For simple models of fixed shape, a fast strategy for image search is to compute the model fit at all possible locations in the image. Pentland et al [11] describe such a fixed-shape model (an eigentemplate) of faces. Their original scheme for face detection measures the residual after fitting the model at each location. Locations where the magnitude of the residual is minimised correspond to possible instances. Cootes et al [2] extend this to account for the probability distribution within the model space, and describe a maximum likelihood-based object detection scheme. Moghaddam and Pentland [9] describe the same approach and show how it can be used for rapid face detection.

The correlation-based methods described above require the model to be of fixed shape. An AAM contains a complete model of appearance, including both shape and texture, and is thus much more specific than an eigentemplate, but is not suitable for direct correlation-based matching. We propose a hierarchical method of using the AAM to rapidly detect objects in images.

### 4.1 Learning the Detection Range

AAM search will converge from a large number of different starting positions. For example, in the case of the face model shown, if at least two-thirds of the model instance at the initial placement overlaps the
true face, the search will converge. In cases where the initial placement is smaller than the true face, the same condition applies, but the model needs to be at least half the size of the face. In every observed case of successful AAM search, the magnitude of the residual between model and image, $|\delta v|^2$, decreases with the first iteration. We can thus provide an initial estimate of possible face locations by running one iteration of AAM search over an equally spaced grid of locations and measuring the residual magnitude. The grid spacing is the widest possible that guarantees at least one starting position sufficiently overlapping the true object.

We will assume an instance might be a plausible place to try further iterations of the AAM if the following condition is met:

$$|\delta v|^2 < T_1$$

where $T_1$ is a pre-learnt threshold. Clearly, the choice of $T_1$ is important. Too low and we risk missing faces - too high and we will detect many false positives. A false positive will cost further wasted iterations and thus processing time.

We can use the training images used to build the model to estimate a suitable value of $T_1$ in advance. Given the known positions and model parameters for the training images, we measure the value of $|\delta v|^2$ after one iteration of AAM search from several displaced starting locations, $c_i$ for each image (again, for simplicity of notation $c_i$ includes scale, angle and translation of the model). In each case we also record whether full AAM search finds the solution. A ‘final-fit’ threshold, $T_f$, defines whether the solution is found. A reasonable choice for $T_f$ is the maximum value of $|\delta v|^2$ for the known ‘best-fit’ to the training examples.

Figure 2 shows an example for our face model, illustrating the effect of varying $T_1$. The y-axis shows the percentage of possible locations that would have been rejected, even though a full search would have converged on the face. The x-axis shows the percentage of wasted searches - when a full search would be requested but would not converge on the face.

In this case, to be sure not to reject possible convergent searches, we must tolerate 55% wasted searches. The key point is that there exists a threshold above which we are guaranteed to keep all convergent searches.

### 4.2 Hierarchical Search Algorithm

The threshold level, $T_1$, can be chosen to include all possible convergent starting locations. Some of these (about 55% in this case) will not be successful and will start to diverge after further iterations. The key is to reject these as soon as possible. The search algorithm is as follows:

1. Perform one iteration at initial grid of $n$ locations, \( \{c_1, c_2, c_n\} \).
2. Measure $|\delta v|^2$ at updated locations $\{c'_1, c'_2, c'_n\}$.
3. Reject instances where $|\delta v|^2 < T_1$.
4. Add instances where $|\delta v|^2 < T_f$ to ‘solutions’ list.
5. Perform further iteration on $m$ remaining instances, $\{c'_1, c'_2, c'_m\}$.
6. Return to 2 until no instances remain.

Thus, as the algorithm iterates, solutions which fall below the final fit threshold, $T_f$, are accepted as plausible objects, whilst solutions which show divergence beyond $T_1$ are rejected. The final list of solutions should describe both the location and model parameters of all objects (face, in this case) in the image.

### 5 Experiments

We tested the system on 200 unseen images containing faces. The first 100 images were fairly straightforward, containing a single face. The second 100 images contained faces hand-segmented from test images and placed in a new image, in order to produce multiple faces on difficult, cluttered backgrounds. We also
tested the system on the same 200 images with the addition of random occlusion. Some typical test images are shown in Figure 6.

5.1 Results
In Table 1 we show the detection rates for each of the image types. The best performance was obtained on the simple, single-face images. This type of image was not significantly affected by occlusion. In the images with multiple faces and cluttered background, occlusion was found to have a small detrimental effect on performance. The system did not return any false-positive matches - the specificity of the model ensures that a low value of $|\delta V|^2$, must correspond to an instance of a face.

<table>
<thead>
<tr>
<th>% of Faces Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
</tr>
<tr>
<td>No Occlusion</td>
</tr>
<tr>
<td>94%</td>
</tr>
</tbody>
</table>

Table 1: Detection rates for different types of test image.

5.2 Search Speed
The time required to search an image depends on the rate at which the hierarchy of possible model instances, $\{c_1, c_2, c_n\}$, is trimmed - images containing multiple faces generally take longer to search. The range of times required to successfully search a 640x480 image was 10 to 35 seconds on a single processor Pentium II 430 MHz machine.

6 Conclusions
In this paper we have extended the Active Appearance Model framework to include coloured texture and to provide the ability to search in occluded images. We have described an iterative hierarchical method for searching entire images using Active Appearance Models. The method is feasible because of the extremely high specificity of Appearance Models. Since the model can only represent legal examples, we can state the image search problem simply:

Find any regions in the image which the model can reconstruct accurately - these must be valid instances.

The resulting system represents a complete and unified scheme for model-based image interpretation, requiring no hand-initialisation. We have applied the approach to the problem of detecting faces in images - a difficult problem due to the large amount of variability. The system was found to locate faces quickly and reliably in the presence of both clutter and occlusion, without generating any false positive matches. We anticipate that such systems will be widely applicable in many areas of image interpretation.

References


