Hierarchical Morphable Models

Michael J. Jones  
Digital Equipment Corporation  
Cambridge Research Lab  
One Kendall Sq., Bldg 700  
Cambridge, MA 02139  
mjones@crl.dec.com

Tomaso Poggio  
Department of Brain and Cognitive Sciences  
Massachusetts Institute of Technology  
E25-201, 45 Carleton Street  
Cambridge, MA 02142  
tp@ai.mit.edu

Abstract

This paper presents a new technique for modelling object classes (such as faces) and matching the model to novel images from the object class. The technique can be used for a variety of image analysis applications including face recognition, object verification and facial expression analysis. The model, called a hierarchical morphable model, is “learned” from example images (partitioned into components) and their correspondences. This is an extension to the work on morphable models described in previous papers ([6], [5], [12]). Hierarchical morphable models are shown to find good matches to novel face images and are also robust to partial occlusion.

1 Introduction

This paper describes research on a component-based extension to morphable models which have been previously described in [6], [5], [12] and [4]. The standard morphable model described in previous papers serves as an image-based method for representing and matching classes of objects (such as faces). Once a morphable model is created for some object class, a novel image from that object class can be matched by the model. The resulting model parameters can be used for a variety of image analysis tasks such as object verification, face recognition and facial expression analysis. One of the problems with morphable models is the number of example images needed to describe an object class. This paper explores a solution to this problem which is to split the example images into components which are less complex and therefore require fewer example images to describe them. For a model of faces, these components would be the eyes, nose and mouth. Furthermore, for very complex objects, components could be decomposed into smaller components to create a hierarchy. We call the new model based on components a hierarchical morphable model. This extension also makes the model more robust to matching partially occluded novel images.

A few other groups have also worked on using shape and texture examples to build models of object classes. Choi et al. [1] for example, studied modelling faces using example shapes and textures. In contrast to our approach, they used a 3-D head model to help find correspondences among the example faces and to novel faces which removed the need for a nonlinear optimization procedure. Cootes et al. ([2], [3]) have developed a model known as an active shape model to represent object classes which is similar in spirit to ours. Another related approach is that of Nastar et al. [7] who treat an image as a deformable 3-D mesh and then use prototypical warps to constrain how the 3-D mesh can be deformed for a particular class. Decomposing a face into components has been proposed by Pentland et al. [8] in their eigenfeature approach. Also, the work of Seitz and Lang [11] explored building object detectors from a hierarchy of components. He used local orientation filters to find evidence for low level features (such as line segments and corners) and then used templates describing the geometric relationships among the features to detect an object.

Before going into more detail about hierarchical morphable models and their advantages, we will first give a brief summary of standard morphable models.

2 Summary of morphable models

The morphable model representation for an object class is “learned” from a set of prototypical images. Each of the prototype grey level images is first “vectorized”, meaning that each image is converted into a shape vector and a texture vector. The shape vector for a prototype is a flow field which contains the pixelwise correspondences to that prototype from a reference prototype. The texture vector for a prototype image is the image in which the grey levels of the image are moved to the corresponding positions in the reference prototype. The shape vector describes
More formally the morphable model is defined as the set of images $I_{model}$, parameterized by 

$$b = [b_0, b_1, \ldots, b_N], c = [c_0, c_1, \ldots, c_N], \text{ and } p = [p_0, p_1, \ldots, p_5],$$

such that

$$I_{model} \circ (A \circ \sum_{i=0}^{N} c_i S_i) = \sum_{j=0}^{N} b_j T_j$$

(2)

where $A: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is the 2D affine transformation

$$A(x, y) = (p_0 x + p_1 y + p_2, p_3 x + p_4 y + p_5).$$

(3)

The summation $\sum_{i=0}^{N} c_i S_i$ describes the shape of every model image as a linear combination of the prototype shapes. Similarly, the summation $\sum_{j=0}^{N} b_j T_j$ describes the texture of every model image as a linear combination of the prototype textures. The global affine transformation, $A(x, y)$, allows the model to represent translations, rotations, scaling and shearing.

A model image for a particular setting of the parameters ($c_i$'s, $b_j$'s and $p_k$'s) is rendered by first computing the linear combination of textures to yield an image which has the same shape as the reference image. This image is then warped according to the flow field obtained by the linear combination of shape vectors and the affine transformation. The warping operation may leave holes in the model image which can be filled by simply interpolating among neighboring pixels. More details about warping can be found in [14].

A morphable model is matched to a novel image by minimizing an error function which directly compares grey level values between the model and novel images. This procedure is described for the hierarchical model in section 3.3.

3 Hierarchical morphable models

Figure 2 illustrates a hierarchical morphable model for a face with 2 levels to the hierarchy. The bottom level contains individual morphable models for the two eyes, the nose and the mouth. The next level of the hierarchy consists of linear combinations of the positions of the centers of each component in the example images. Information about valid facial configurations is represented by these positions. The output of the model uses these linear combinations of vectorized components and positions to synthesize an image containing all the components in the appropriate positions.

A major advantage of a hierarchical model (relative to the standard morphable model) is that fewer prototypes are necessary for each component to achieve the same quality of matches. This is because each
Morphable models

Figure 2: An illustration of a hierarchical morphable model with 2 levels. The bottom level consists of component morphable models for the eyes, nose and mouth. The next level consists of linear combinations of the positions of each component. A particular parameter setting yields an output face built from components.

Linear combination of positions:

Output of hierarchical model:

A particular parameter setting yields an output face built from components.

Component will be less complex than the whole image and thus require fewer examples to span the possible component images. The model is also more expressive since there are more degrees of freedom. One can think of it as a piecewise linear model instead of just a linear model. Another advantage is that occlusion can be handled better using components. For example, if one component is occluded in the novel image then the other components can still be matched without being negatively affected by the occluded part of the image. With the standard morphable model, occluded images can be matched, but the quality of the match degrades with increasing amounts of occlusion.

3.1 Manually specifying components

We are not currently addressing the problem of automatically finding components in a set of prototypes. Instead we have written a simple program which allows us to manually specify the components in each prototype by placing rectangles on each prototype to specify the region for each component. Figure 3 illustrates this process. Manually specifying components is not an unreasonable burden since it is only necessary in the model building phase. After this phase, the components of a novel input image are automatically found by the matching algorithm.

3.2 Formal specification

The formal specification for hierarchical morphable models is similar to standard morphable models with the addition of parameters which control the position of each component.

As before we have a set of prototype images $I_0, \ldots, I_N$. The reference image is $I_0$. The components are manually specified on $I_0$ by rectangular regions. The center of the rectangle along with the height and width give the size and position of each component. Let $K$ be the number of components in each prototype.

Let $C_k$ be the set of $x,y$ coordinates of $I_0$ which are contained in component $k$, i.e. $C_k$ is a rectangular region of $I_0$ defining component $k$.

For each prototype $i$, the $k$th component has correspondence field

$$S^k_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2.$$  

$S^k_i(x,y) = (\hat{x}, \hat{y})$ where $x,y$ is a pixel in component $k$ of the reference image and $\hat{x}, \hat{y}$ is the corresponding point in component $k$ of prototype $i$. Note that there is a different set of shape vectors for each different component.

Similarly, $T^k_i$ is the texture vector of component $k$ for prototype $i$.

$$T^k_i(x,y) = I_i \circ S^k_i(x,y) \quad \forall (x,y) \in C_k.$$
We add a new vector, \( P_i^k \), to the model which is the displacement of the \((x, y)\) position of the center of component \(k\) in prototype \(i\) (relative to its position in the reference image). In other words, \( P_i^k \) is simply a vector pointing from the center of component \(k\) in \(I_0\) to the center of component \(k\) in \(I_i\).

\[
P_i^k = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}.
\]

As with shape and texture, the position of each component in any model image is constrained to be a linear combination of prototype component positions. The function \(Q^k : \mathbb{R}^2 \rightarrow \mathbb{R}^2\) describes this linear combination and is defined as

\[
Q^k(x, y) = \sum_{i=1}^{M} a_i P_i^k + \begin{bmatrix} x \\ y \end{bmatrix}
\]

where \(M\) is the number of prototype positions used. \(M\) is typically less then \(N\) (the number of prototypes) since the position vector space is much lower dimensional than the shape or texture space. The position vector space has dimensionality \(2 \times K\) since each of the \(K\) components has 2 elements \((x\) and \(y)\) in its position vector.

Next, we would like each component to have affine transformations independently as well as having a global affine transformation. This will allow an individual mouth component, for example, to rotate independently of the eyes and nose and at the same time allow the positions of all the components to translate, rotate or scale using the global affine parameters. To do this, we will add a component affine transformation, \(A^k\), as defined before to each component model and a global affine mapping, \(B\), to the position function, \(Q^k\). Including translation parameters in both affine transformations is redundant so we exclude them from the component affine transformations since the relative positions of the components should be constrained by the linear combination of prototype positions. Let

\[
A^k(x, y) = (p_0^k x + p_1^k y + p_2^k, p_3^k x + p_4^k y + p_5^k)
\]

and

\[
B(x, y) = (e_0 x + e_1 y + e_2, e_3 x + e_4 y + e_5).
\]

Now we can define the hierarchical morphable model as the set of all images \(I_{\text{model}}\) such that for each component \(k\) and \((x, y) \in C_k\)

\[
I_{\text{model}} = B \circ Q^k \circ (A^k \circ \sum_{i=0}^{N} c_i^k S_i^k)(x, y) = \sum_{j=0}^{N} b_j^k T_j^k(x, y).
\]

The mapping \((A^k \circ \sum_{i=0}^{N} c_i^k S_i^k)\) constrains the shape of component \(k\) to be a linear combination of prototype component shapes followed by a local affine transformation (excluding translation). The mapping \(B \circ Q^k\) constrains the position of component \(k\) to be a linear combination of prototype component positions followed by a global affine transformation. The mapping \(\sum_{j=0}^{N} b_j^k T_j^k\) constrains the texture of component \(k\) to be a linear combination of prototype component textures.

The parameters of the hierarchical morphable model are the shape parameters, \(c^k\), the texture parameters, \(b^k\), the position parameters \(a\), the component-level affine parameters \(p^k\) and the global affine parameters \(e\). The index \(k\) goes from 1 to \(K\) (the number of components).

Similarly to the standard morphable model, a model image in the hierarchical model is rendered by warping the linear combination of textures according to the linear combination of shapes plus component translation (from the linear combination of position vectors) and affine transformations. The warping is done for each component individually.

### 3.3 Matching the hierarchical model

To match a hierarchical morphable model to a novel image, we first define an error function between the novel image and the current guess for the best fitting model image.

We define the sum of squared differences error as follows:

\[
E(c, b, a, p, e) = \frac{1}{2} \sum_{k=1}^{K} \sum_{(x, y) \in C_k} \left[ I_{\text{novel}}(x, y) - I_{\text{model}}(x, y) \right]^2
\]

where \(I_{\text{novel}}\) is the novel gray level image being matched and \(I_{\text{model}}\) is the current guess for the best fitting model image. By applying the shape transformation (given given estimated values for the model parameters) to \(I_{\text{novel}}\) and \(I_{\text{model}}\) and using equation 7 we get

\[
E(c, b, a, p, e) =
\]

\[
\frac{1}{2} \sum_{k=1}^{K} \sum_{(x, y) \in C_k} \left[ I_{\text{novel}} \circ B \circ Q^k \circ (A^k \circ \sum_{i=0}^{N} c_i^k S_i^k)(x, y) - \sum_{j=0}^{N} b_j^k T_j^k(x, y) \right]^2.
\]

Minimizing this error with respect to the model parameters yields the model image whose components best fit the novel image with respect to the \(L_2\) norm. Stochastic gradient descent ([13], [10]) is used to do
Figure 4: The face prototypes were split up into 4 components. 50 prototypes were used for each component, 10 of which are shown here.

Figure 5: Six example matches of the standard morphable face model to novel input images.

this minimization. Due to space limitations, the necessary derivatives are not given here, but can be computed straightforwardly using the chain rule. Stochastic gradient descent is faster than standard gradient descent because it estimates the derivatives by sampling a small number of pixels in each component rather than using every single pixel in each component. This is described in more detail in [6].

The model parameters are typically initialized to 0 (except for the affine transformations which are initialized to the identity transformation). Furthermore, it is assumed that the position of the initial model is roughly on top of the target object in the novel image.

4 Example hierarchical model

As an example of a hierarchical morphable model we will use faces split into eyes, nose and mouth components as shown in figure 2. We use as prototypes the face database for which a subset is shown in figure 1. The components were selected manually as described in section 3.1. Fifty prototypes were used for each component. We found that 50 prototypes were enough to get very good matches. Figure 4 shows 10 of the 50 prototypes used to define each component.

The correspondence fields for each component were taken from the full image correspondence fields automatically computed for these faces using the bootstrapping algorithm described in [12].

Because we are using four components in this hierarchical morphable model, there are 8 elements in the “position vector space”. The position prototypes ($P^k_i$) have 2 elements ($\Delta x, \Delta y$) for each of the four components. Therefore, 8 linearly independent position prototypes for each component would completely span the position vector space, effectively making the positions of the components unconstrained. We used the first 7 eigenvectors of the position vector space to constrain the component positions.

The stochastic gradient descent algorithm was run using 15 samples per iteration and 7000 iterations. The running time was about 40 seconds on a 333 Mhz DEC Alpha.

The results of matching the hierarchical morphable model to some novel input images are shown in figure 6. There are some edge artifacts around each component due to the warping algorithm used to render these images. However, by comparing these examples to the results on the standard morphable model shown in figure 5 we see that the hierarchical model finds better matches for each component. The standard morphable model was built from 100 face prototypes - twice as many as were used by the hierarchical model. Due to the low resolution of the figure the improved match quality may be difficult to see.

This hierarchical morphable model was also tested on its ability to match occluded images. Figure 7 shows some examples of matching novel images which are occluded by solid rectangles. The matches are good for those components which do not contain occlusion. The reconstruction, however, of components which are even partially occluded is poor. This is the
main difference between the performance of the standard morphable model and the hierarchical morphable model on occluded images. The hierarchical model does a good job of matching unoccluded components even though the rest of the image may be heavily occluded. The quality of matches for the standard model degrades even in areas where there is no occlusion. On the other hand, the standard model does a better job of reconstructing areas where there is partial occlusion because it uses information from the entire image to find a match.

5 Summary and Conclusions

We have presented an extension of morphable models to a hierarchical model of components. We have argued that this extension requires fewer example prototypes than the standard model and that it is more expressive even with fewer examples. We have demonstrated these points with an example using a model of faces.

More complex models than the one shown here could be created by adding more layers to the hierarchy. In other words, higher level components could themselves contain hierarchical models.

So far we have only mentioned specific applications briefly. Rihert and Jones [9] present an application of morphable models to gaze estimation.

Another interesting application for hierarchical morphable models is object verification. Such a model may be especially useful if the object can be occluded. In this case a heuristic can be used such as “at least two components must have a good match” in order to report a successful detection.

For face recognition, the model parameters after matching a particular face could be used as input to a classifier to determine the person’s identity.

For more discussion of applications see [6].

Hierarchical morphable models provide an intriguing framework for using example images to build models of object classes. The parameters of the model after matching provide a higher level representation of the image contents that can be used for many image analysis problems.

References

Figure 7: Six examples of matching a partially occluded face using the hierarchical face model. Unoccluded components are matched well independently of occluded regions.


