RECOGNIZING FACES FROM A NEW VIEWPOINT

Thomas Vetter

Max-Planck-Institut für biologische Kybernetik
Spemannstr. 38 72076 Tübingen – Germany
E-mail: vetter@mpik-tueb.mpg.de

ABSTRACT

A new technique is described for recognizing faces from new viewpoints. From a single 2D image of a face synthetic images from new viewpoints are generated and compared to stored views. A novel 2D image of a face can be computed without knowledge about the 3D structure of the head. The technique draws on prior knowledge of faces based on example images of other faces seen in different poses and on a single generic 3D model of a human head. The example images are used to learn a pose-invariant shape and texture description of a new face. The 3D model is used to solve the correspondence problem between images showing faces in different poses. The performance of the technique is tested on a date set of 200 faces of known orientation for rotations up to 90°.

1. INTRODUCTION

In recent papers [7] we introduced a technique that allows to generate new views of a face from a single image. With viewpoint changes, some previously visible regions of the object become occluded, while other previously invisible regions become visible. Additionally, the configuration of object regions that are visible in both views may change. Accordingly, to synthesize a novel view of an object, two problems must be resolved. First, the visible regions that the new view shares with the previous view must be redrawn at their new positions. Second, regions not previously visible from the view of the example image must be generated or synthesized. It is obvious that this latter problem is unsolvable without prior assumptions.
In recent years, two-dimensional image-based face models have been applied for the synthesis of rigid and non-rigid face transitions [3, 6]. These models exploit prior knowledge from example images of prototypical faces and work by building flexible image-based representations (*active shape models*) of known objects by a linear combination of labeled examples. The underlying coding of an image of a new object or face is based on linear combinations of the two-dimensional shape of examples of prototypical images. A similar method has been used to synthesize new images of a face with a different expression or a changed viewpoint [3] making use of only a single given image. The most serious limitation of this techniques is their reliance on the solution of the correspondence problem across view changes. Over large changes in viewpoint, this is still highly problematic due to the frequency with which occlusions and occluding contours occur.
To overcome these difficulties in the present work, we draw on the concept of *linear object classes*, which we have introduced recently in the context of object representations [8]. This approach does not need correspondence across different viewpoints and therefore is capable of coping with larger viewpoint changes. For each specific viewpoint a separate linear image model is used where each leads to the same view independent representation of an object. An image is coded by one linear model and this code is used in a different linear model to synthesize the new view. While this basic coding scheme is advantageous for handling large viewpoint changes, however, it has some drawbacks for information not representable by the linear coding model. For instance textural details, like moles and blemishes, will be lost with this linear modeling approach, even when they are clearly visible in the given image. However, combining the linear object class approach with a single 3D model of a human head will retain all its advantages without the loss of textural information [7].

For face recognition, this image synthesis technique allows to transform a given image of a face into a new image, depicting the face in a standard orientation in which a view of the face might be stored.

2. ALGORITHM FOR VIEW SYNTHESIS

In this section the algorithm is described that allows for the synthesis of novel views of a face from a single example view of the face [7]. For brevity, in the present paper we describe the application of the algorithm to

the synthesis of a "frontal" view from an example "rotated" view. It should be noted, however, that the algorithm is not at all restricted to a particular orientation of faces.
The algorithm can be subdivided into three parts.

- First, the texture and 2D-shape information in an image of a face are separated.
- Second, two separate modules, one for texture and one for 2D-shape, compute the texture and shape representations of a given "rotated" view of a face. These modules are then used to compute the 2D-shape and texture estimates for the new "frontal" view of that face.
- Finally the new texture and 2D-shape for a "frontal" view are combined and warped to the "frontal" image of the face.

Separation of texture and 2D-shape in images of faces: The central part of the approach is a representation of face images that consists of a separate texture vector and 2D-shape vector, each one with components referring to the same feature points – in this case pixels. Assuming correspondence the 2D-shape of a face image is represented by a vector $\mathbf{s} = (x_1, y_1, x_2, ..., x_n, y_n)^T \in \Re^{2n}$, that is by the x, y displacement of each feature with respect to the corresponding feature in the reference face. The texture is coded as a difference map between the image intensities of the exemplar face and its *corresponding* intensities in the reference face. Such a normalized texture can be written as a vector $\mathbf{T} = (i_1, ...i_n)^T \in \Re^n$, that contains the image intensity differences i of the n pixels of the image. All images of the training set are mapped onto the reference face of the corresponding orientation. This is done separately for each rotated orientation. Automated procedures for this nonlinear normalization are found in the optical flow literature and its application to faces [1, 3, 7].

Module for 2D-shape processing: The 2D-shape model of human faces used in the algorithm is based on the linear object class idea (the necessary and sufficient conditions are given in [8]) and is built on a training set of pairs of images of human faces. From each pair of images, each consisting of a "rotated" and a "frontal" view of a face, the 2D-shape vectors s^r for the "rotated" shape and s^f for the "frontal"shape are computed. Consider the three-dimensional shape of a human head defined in terms of pointwise features. The 3D-shape of the head can be represented by a vector $\mathbf{S} = (x_1, y_1, z_1, x_2, ..., y_n, z_n)^T$, that contains the x, y, z-coordinates of its n feature points. Assume that

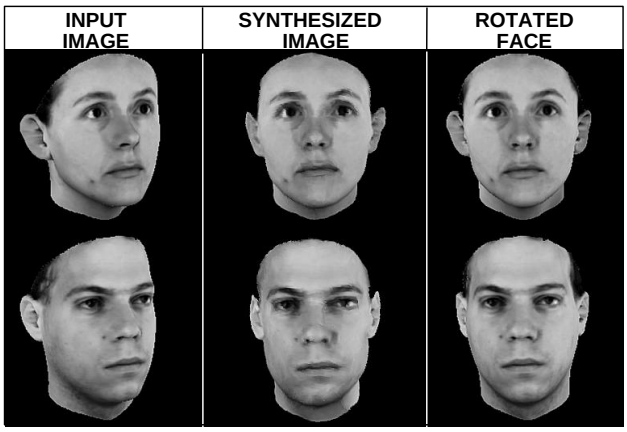

Figure 1: *Synthesized frontal views (center column) to a given rotated* (24°) *image of a face (left column) are shown. The prior knowledge about faces was given through a training set of 99 pairs of images of different faces (not shown) in the two orientations. Additionally a single 3D-head model for the reference face was used to establish correspondence across the view point change. The frontal image of the real face is shown in the right column.*

$\mathbf{S} \in \Re^{3n}$ is the linear combination of q 3D shapes $\mathbf{S}_i$ of *other* heads, such that: $\mathbf{S} = \sum_{i=1}^{q} \beta_i \mathbf{S}_i$. It is quite obvious that for any linear transformation R (e.g. rotation in 3D) with $\mathbf{S}^r = R\mathbf{S}$, it follows that $\mathbf{S}^r = \sum_{i=1}^{q} \beta_i \mathbf{S}_i^r$. Thus, if a 3D head shape can be represented as the weighted sum of the shapes of other heads, its rotated shape is a linear combination of the rotated shapes of the other heads with the same weights β_i.
To apply this to the 2D face shapes computed from images, we have to consider the following. A projection P from $3D$ to $2D$ with $\mathbf{s}^r = P\mathbf{S}^r$ under which the minimal number q of 2D-shape vectors necessary to represent $\mathbf{S}^r = \sum_{i=1}^{q} \beta_i \mathbf{S}_i^r$ and $\mathbf{s}^r = \sum_{i=1}^{q} \beta_i \mathbf{s}_i^r$ does not change, it allows the correct evaluation of the coefficients β_i from the images. Or in other words, the dimension of a three-dimensional linear 2D-shape class is not allowed to change under a projection P. Assuming such a projection, and that s^r, a 2D shape of a given "rotated" view, can be represented by the "rotated" shapes of the example set s_i^r as

$$s^r = \sum_{i=1}^{q} \beta_i \mathbf{s}_i^r, \qquad (1)$$

then the "frontal" 2D-shape $\mathbf{s}^f$ to a given $\mathbf{s}^r$ can be computed without knowing $\mathbf{S}$ using β_i of equation (1) and the other $\mathbf{s}_i^f$ given through the images in the training set with the following equation:

$$\mathbf{s}^f = \sum_{i=1}^{q} \beta_i \mathbf{s}_i^f. \qquad (2)$$

In other words, a new 2D face shape can be computed without knowing its three-dimensional structure. It should be noted that no knowledge of correspondence between equation (1) and equation (2) is necessary (rows in a linear equation system can be exchanged freely).

Module for texture processing: In contrast to the shape model, two different possibilities for generating a "frontal" texture given a "rotated" texture are described. The first method is again based on the linear object class approach and the second method uses a single three-dimensional head model to map the texture from the "rotated" texture onto the "frontal" texture. The linear object class approach for the texture vectors is equivalent to the method described earlier for the 2D-shape vectors. It is assumed that a "rotated" texture $\mathbf{t}^r$ can be represented by the q "rotated" textures $\mathbf{t}_i^r$ computed from the given example set as follows: $\mathbf{t}^r = \sum_{i=1}^{q} \alpha_i \mathbf{t}_i^r$. The new texture $\mathbf{t}^f$ is generated by combining the "frontal" example textures using the computed weights α_i as follows $\mathbf{t}^f = \sum_{i=1}^{q} \alpha_i \mathbf{t}_i^f$.

A single 3D head model: Whereas the linear texture approach is satisfactory for generating new "frontal" textures for regions not visible in the "rotated" texture, it is not satisfactory for the regions visible in both views. The linear texture approach is hardly able to capture or represent features which are particular to an individual face (e.g. freckles, moles or any similar distinct aspect of facial texture). Such features ask for a direct mapping from the given "rotated" texture onto the new "frontal" texture. However, this requires pixelwise correspondence between the two views (see [3]).
Since all textures are mapped onto the reference face, it is sufficient to solve the correspondence problem across the the viewpoint change for the reference face only. A three-dimensional model of an object intrinsically allows the exact computation of a correspondence field between images of the object from different viewpoints, because the three-dimensional coordinates of the whole object are given, occlusions are not problematic and hence the pixels visible in both images can be separated from the pixels which are only visible from one viewpoint.
Final image synthesis: The texture obtained through direct texture mapping across the viewpoint change and the texture obtained through the linear class approach are merged by standard image blending techniques. This new texture is finally warped along the generated new 2D-shape vector to the new image representing a new view to the input face image.

3. FACE IMAGES AND HEAD MODEL.

Images of 200 caucasian faces, showing views of (0°, 30°, 60° and 90°). The images were originally rendered under mainly ambient illumination conditions from a data base of three-dimensional human head models recorded with a laser scanner ($Cyberware^{TM}$). The simulated pin-hole camera was set to a distance of 120 cm from the face. The different views were taken by moving the camera around the face. The head hair was removed digitally (but with manual editing), via a vertical cut behind the ears. The resolution of the grey-level images was 256-by-256 pixels and 8 bit.
Preprocessing: First the faces were segmented from the background and aligned roughly by automatically adjusting them to their two-dimensional centroid. The centroid was computed by evaluating separately the average of all x, y coordinates of the image pixels related to the face independent of their intensity value.
A single three-dimensional model of a human head was used to render the two reference images and to compute the correspondence field between these two images. This model was the average of 50 three-dimensional models of human heads, recorded with a laser scanner ($Cyberware^{TM}$).

4. RESULTS

The algorithm was tested on 200 human faces. For each face, images were given in four orientations (0°, 30°, 60° and 90°).

The data was divided into two subsets of 100 faces each. The two sets were used as a test and a training set and vice versa. For each orientation separately a linear model was built from the 100 faces of the training set. For each orientation and each face of the test set, new views were generated using the linear models obtained from the training set and the synthesis technique described earlier.

For each synthetic image, the most similar image in the whole data set of 200 different faces in the same orientation was computed. For comparison the *Euclidean distance* (L_2) was applied to the images in pixel representation without further processing.

The error rates evaluated over all 200 faces are plotted in figure 2. For rotations of 30° the error rate was at maximum 2%. For rotations of 60° the error rate was on average 17.5% and for 90° the error reached 45%. Pure chance would have lead to an error rate of 99.5%

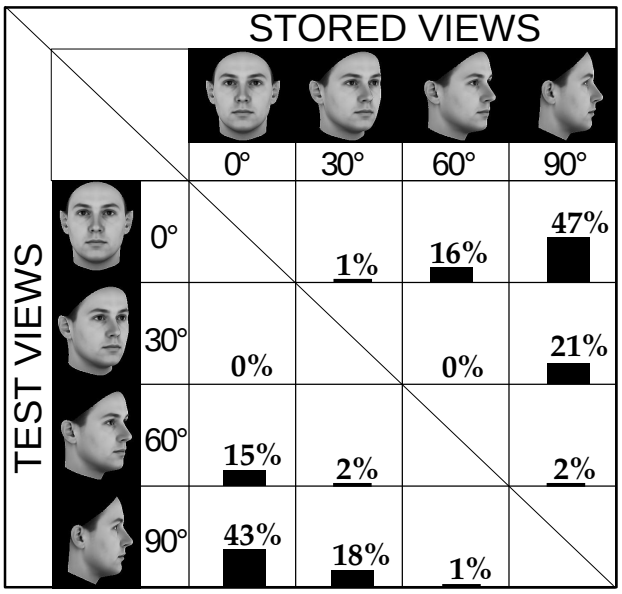

Figure 2: *Recognition errors evaluated on 200 faces for each transfer condition. For each test view new views were generated and compared to 200 images showing different faces in the same orientation. The model for the view synthesis was built on 100 training faces.*

5. DISCUSSION

The results indicate the importance of the proposed face model for viewpoint independent face recognition systems.

Several open questions remain for a fully automated implementation. The key step in the proposed technique is a correspondence field between images of faces seen from the same viewpoint. The optical flow technique used worked well, however, for images obtained under less controlled conditions, a more sophisticated method for finding the correspondence might be necessary. The use of images derived from 3D-head models allowed the generation of identical illumination conditions for all example and test images. An extension of the proposed method to test images obtained by a normal camera will lead to more unconstrained lighting conditions and will influence the correspondence finding step. Both problems have been investigated by Hallinan (1995) for frontal face images. By fitting a linear model to an image, he could determine the lighting conditions as well as the correspondence. This approach is very similar to the correspondence techniques based on *active shape models* [6, 5], which are more robust to local occlusions when applied to a known object class.

One of the most critical assumptions in the method presented here, is that the orientation of a face in the image must be known. Different techniques have been reported to estimate the orientation of faces [2, 6] . The approach [6] using a flexible face model for pose estimation was precise to $2 - 5°$. It is not clear yet how precisely the orientation should be estimated to yield satisfactory results. However, considering that there was still some variance in the pose of the faces in our data set, a precision of 2° seems promising.

6. REFERENCES

[1] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 237–252, Santa Margherita Ligure, Italy, 1992.

[2] D. Beymer and T. Poggio. Image representation for visual learning. *Science*, 272:1905–1909, 1996.

[3] D. Beymer, A. Shashua, and T. Poggio. Example-based image analysis and synthesis. A.I. Memo No. 1431, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[4] P.W. Hallinan. A deformable model for the recognition of human faces under arbitrary illumination. Doctoral thesis, Harvard University, Cambridge, Massachusetts, 1995.

[5] M. Jones and T. Poggio. Model-based matching of line drawings by linear combination of prototypes. In *Proceedings of the 5th International Conference on Computer Vision*, 1995.

[6] A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmad. Automatic interpretation of human faces and hand gestures using flexible models. In M.Bichsel, editor, *Proc. International Workshop on Face and Gesture Recognition*, pages 98–103, Zurich, Switzerland, 1995.

[7] T. Vetter. Learning novel views to a single face image. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 22–27, Killington, VT, 1996.

[8] T. Vetter and T. Poggio. Image synthesis from a single example image. In B. Buxton and R. Cippola, editors, *Computer Vision – ECCV'96*, Cambridge UK, 1996. Springer, Lecture Notes in Computer Science 1065.