Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)
14-16 April 1998, Nara, Japan.

Real-Time Face Recognition Using Feature Combination

Chahab Nastar & Matthias Mitschke
INRIA
B.P. 105
78153 Le Chesnay Cedex, France
Chahab.Nastar@inria.fr

Abstract

We present an experimental setup for real-time face identification in a cluttered scene. Color images of people are recorded with a static camera. A rough face detection is performed, and the resulting images are stored in a database. At a future time, a person standing in front of the camera (although against a different background) is identified, if their image was present in the database. In our experiments, the main variation of the faces is wide pose variation (out-of-image plane rotation of the head); some scale variation was also present. For real-time ability, we combine simple image features through a voting procedure for performing face recognition.

1. Introduction

Face Recognition has been studied for over 20 years in computer vision [6, 7]. Since the beginning of the 1990s, the subject has become a major issue, mainly due to the important real-world applications of face recognition: smart surveillance, secure access, telecommunications, digital libraries, medicine… (a survey can be found in [2]).

On the theoretical side, face recognition is a specific and hard case of object recognition. Faces are very specific objects whose most common appearance (frontal faces) roughly look alike. Subtle changes make the faces different. In a traditional feature space, frontal faces will form a dense cluster, and standard pattern recognition techniques will generally fail to discriminate between them.

Therefore, face recognition needs dedicated representations of the images and specific algorithms. Successful approaches include eigenfaces [8, 20, 14], neural nets [15, 4], natural basis functions [16], template matching [1], active shape models [9], discriminant analysis [19], and flexible images [12, 10] (the reader is refered to the FG'95 and FG'96 proceedings for a good overview of recent techniques).

Unfortunately, most of these techniques are complex and cannot match the real-time constraint with the computers that we have today. The eigenface approach is one of the few techniques that can be made real-time if the scale and the pose of the faces do not vary too much [17] ; nevertheless, it would need several eigenspaces (for each face pose), which would be rather complex to deal with.

In this paper, we investigate the use of simple and easy-to-compute image features whose combination will lead to a good recognition performance. Our motivation is twofold: (i) we are interested in the real-time ability of simple image features (ii) we want to prove experimentally that a smart combination of low-level and generic image features can lead to identification of hard and specific objects such as faces.

The identification takes place in 2 steps and operates on color images :

1. Face detection and background masking. This stage is dealt with by the MIT Pfinder (person-finder) software [21]. It is detailed in section 2.

2. Face idenfication. This stage is viewed as an image retrieval problem and uses a voting procedure. It is detailed in section 3.

Our experimental setup is simple: a cheap color video camera is plugged into a Maximum Impact SGI machine (with real-time digitizing card). The camera is pointing to a complex background. A person walks in the scene, several of their pictures are captured and compared to a previously recorded face database for identification.

2 Face detection

The face detection stage is based on the Pfinder (person finder) software[1]. Pfinder is a real-time system for tracking silhouettes. The system uses a multi-class statistical model

[1] From MIT Media Lab, Vision and Modeling group.

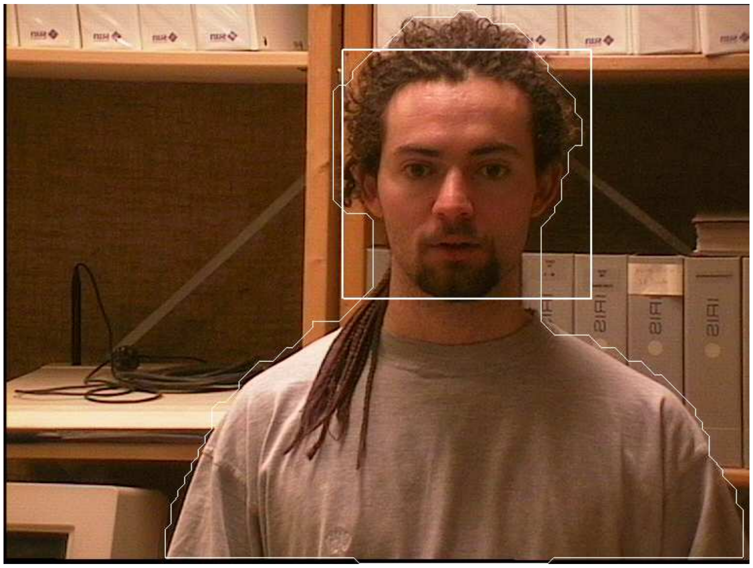

Figure 1. Finding the person's silhouette using Pfinder.

of color and shape to obtain a 2D representation of the silhouette in a wide range of viewing conditions. Pfinder has performed reliably on thousands of people in many different physical locations [21].

Technically, the output of Pfinder is the silhouette of the person in the scene. Obviously, Pfinder allows to mask out the background, keeping only the person (figure 1).

Locations of specific landmarks of the silhouette, such as the head and the hands, are also estimated by Pfinder. We use the estimated head location to create the face image. We have investigated several methods for creating the face image :

1. The output image has a predetermined spatial resolution (e.g. 128x128) around the location of the head. This method is only convenient with small scale variations.

2. The output image has a spatial resolution such that the ratio of the number of face pixels to the number of pixels in the image is constant (e.g., 80%). The difficulty with this method is that images will have different spatial resolutions, making their comparison more complex.

This output image (also called "query image", as it will be used for an image database search) can be further corrected. Indeed, given the spatial resolution and the shadowing effects, Pfinder will generally over-estimate the silhouette of the person by a few pixels. By using a simple filtering technique (erosion, from mathematical morphology), we can obtain a corrected image that will more closely match the silhouette of the person.

An example of the full face detection chain is shown in figure 2.

3 Face Identification

We now can perform face identification. This problem is addressed as an image database search. More precisely, the input image is going to be analyzed on-line, and the result of the analysis will be compared to the results obtained with the images in the database. We use our own homebrew face database of people in the lab. The database contains 160 color images (16 people, 10 images each). Each image is labeled with the person's name. The subjects were asked to mainly perform head rotations in front of the camera (see an example on figure 3).

The face identification stage is viewed as a query-by-example image retrieval problem (figure 4). More precisely:

1. c (e.g. $c = 5$) images of the moving face are detected (section 2) and captured on-line.

2. f (e.g. $f = 4$) image features are used to derive image signatures for these c images (section 3.1).

3. the database is looked up for similar images (nearest neighbors according to a metric) and a voting procedure on image features allows to retrieve the best matches for the feature combination, thus determining the identity of the face.

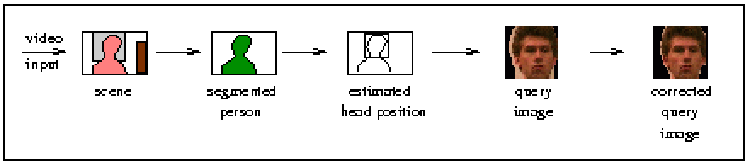

Figure 2. Detecting the face using Pfinder and an erosion technique.

Figure 3. Typical pose variations of the face.

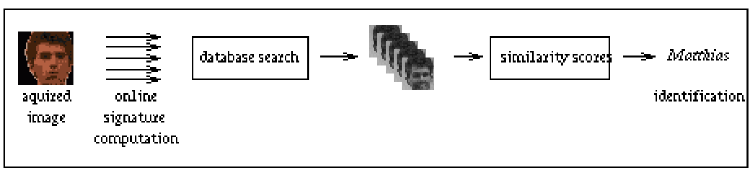

Figure 4. The face identification process.

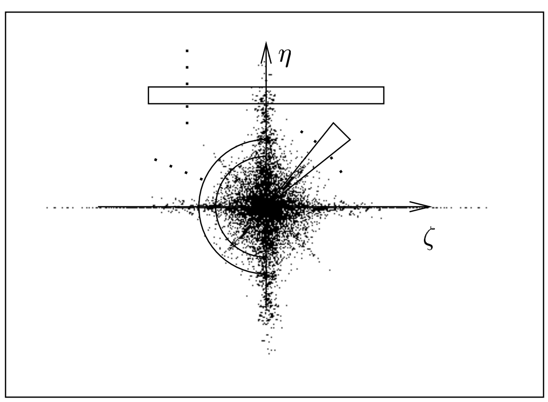

Figure 5. Computation of the texture signature from Fourier power spectrum.

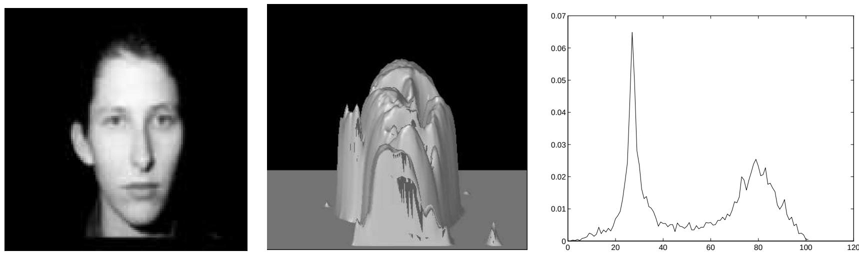

Figure 6. An image, its gray surface, and its image shape spectrum (ISS)

3.1 Image signatures

For real-time applications, wide pose variations (and some scale variation and facial expression), our idea is to use **simple** image features and to **combine** them for face identification, since dedicated face recognition algorithms are usually too intensive. Classically, simple image features include color, texture and shape. We also include the image shape spectrum, a more complex image feature that can be quickly computed on-line and that has proven to be effective for image retrieval [11].

The color feature is accounted for via the 64-bin L1-normalized color histogram in *rgb* space [18, 5]. Although more suited color spaces (such as Luv or Ohta) could be used, the use of the *rgb* space has the advantage of being particularly easy and fast. The L1-normalized color histogram has the important scale-invariance property; as any histogram, it is also translation and rotation invariant.

Texture as a spectral phenomenon can be described by the Fourier power spectrum of the image. After a Fast Fourier Transform and computation of the power spectrum, mean values over bar-, ring-, or wedge-shaped areas of the spatial frequency plane are computed [13] (see figure 5). Note that the ring-based integration offers rotation invariance, whereas the wedge-shaped integration allows for scale invariance. We use the wedge-shaped integration such that our texture signature is scale-invariant, which is the main invariance that we are interested in.

Shape is captured in a global way, using the structure of the edges over the entire image. After applying a Canny edge detector, the histogram of edge directions is computed, providing the orientation histogram [3, 5]. Unconnected edge points and very short edge segments are eliminated by thresholding before the histogramming stage. Normalizing the orientation histograms supply the scale-invariant property.

Finally, the *image shape spectrum (ISS)* is also computed [11]. The ISS is based on differential geometry and is derived as follows: We define the *image shape index* as a quantitative measure of the local shape of the intensity surface at a point $p = (x, y, I(x, y))$:

$$S_I(p) = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1(p) + k_2(p)}{k_1(p) - k_2(p)} \tag{1}$$

where k_1 and k_2 are the principal curvatures of the intensity surface, and $k_1 \geq k_2$. The principal curvatures are derived by using Gaussian filtering. We now build the *image shape spectrum* (ISS) as the histogram of the image shape index over the entire image. We normalize the ISS so that it sums up to unity. An example of an image, its greylevel surface, and its ISS[2] are represented on figure 6.

The ISS has translation, rotation and scale invariance properties, and captures the significant information contained in the data particularly well. It has been successful with image retrieval applications [11].

To sum up, we now have $f = 4$ image signatures: color histogram, Fourier texture, orientation histogram, and image shape spectrum. We now need a similarity metric for comparing image signatures. The simple L_1 metric is fast to compute and well-suited since we are mainly dealing with histograms.

3.2 Voting procedure

Each of the previous signatures will now *vote* for its preferred image (i.e. the image which is most similar to the query image, whose signatures were computed on-line). More precisely, each of the f signatures ranks the images in the database (with decreasing order of similarity). For each image, we sum up its f different rankings and get the number r. We then retrieve R (we choose $R = 15$) images with increasing r.

Now, for each of the c captured images, we can retrieve R images from the database. By counting the number of each different identity and their rankings, we can decide for one single most similar image for each captured image.

[2] The x axis represents discrete values of the shape index, the bin width being unity (101 bins in total); at each bin, the y axis represents the proportion of surface points that have the same shape index.

Again, a simple voting procedure will identify the person standing in front of the camera.

Figure 7 shows an example. On the top row are the $c = 5$ on-line captured images of Richard, on the bottom row the corresponding best matches in the database (using the voting procedure described above). Note that despite one false hit, the system was able to identify Richard in real-time. On figure 8, we show another example where we display the complex background. Note the scale and pose variations between the top row (captured images) and the bottom row (database images).

4. Conclusion

We have presented a real-time face recognition system. The system is based on **simple** image feature computation and their **combination** using a simple voting procedure. The proposed technique allows for real-time ability. The system performed very well on our different experiments. Future work includes testing with more complex face variations (such as occlusion or expression) and dealing with illumination invariance.

Acknowledgements

We wish to thank Prof. Alex Pentland (MIT, Media Lab, Vision and Modeling group) for providing the Pfinder software.
Thanks are also due to Haico Ten Lohuis for partial implementation of the approach described in this paper.

References

[1] D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of the International Conference on Computer Vision 1995 (ICCV'95)*, Cambridge, MA, June 1995.

[2] R. Chellappa, S. Sirohey, C. Wilson, and C. Barnes. Human and machine recognition of faces : A survey. Technical Report CAR-TR-731, CS-TR-3339, University of Maryland, August 1994.

[3] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proceedings of the 1st International Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, 1995.

[4] S. Gutta, J. Huang, I. Imam, and H. Wechsler. Face and gesture recognition using hybrid classifiers. In *International Conference on Automatic Face and Gesture Recognition*, Killington, VT, October 1996.

[5] A. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8), 1996.

[6] T. Kanade. Picture processing by computer complex and recognition of human faces. Technical report, Kyoto University, Dept. of Information Science, 1973.

[7] T. Kanade. *Computer Recognition of Human Faces*. Basel and Stuttgart : Birkhauser, 1977.

[8] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), January 1990.

[9] A. Lanitis, C. J. Taylor, and T. F. Cootes. A unified approach to coding and interpreting face images. In *Proceedings of the International Conference on Computer Vision 1995 (ICCV'95)*, Cambridge, MA, June 1995.

[10] B. Moghaddam, C. Nastar, and A. Pentland. Bayesian face recognition using deformable intensity surfaces. In *IEEE Proceedings of Computer Vision and Pattern Recognition (CVPR '96)*, San Francisco, June 1996.

[11] C. Nastar. The image shape spectrum for image retrieval. Technical Report 3206, INRIA, July 1997. ftp.inria.fr/INRIA/publication/RR/RR-3206.ps.gz.

[12] C. Nastar, B. Moghaddam, and A. Pentland. Flexible images: Matching and recognition using learned deformations. *Computer Vision and Image Understanding*, 35(2), February 1997.

[13] H. Niemann. *Pattern Analysis and Understanding*. Springer, Heidelberg, 1990.

[14] A. Pentland, B. Moghaddam, T. Starner, and M. Turk. View based and modular eigenspaces for face recognition. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, 1994.

[15] T. Poggio and D. Beymer. Learning networks for face analysis and synthesis. In *International Conference on Automatic Face and Gesture Recognition*, Zurich, 1995.

[16] R. P. N. Rao and D. H. Ballard. Natural basis functions and topographic memory for face recognition. In *IJCAI*, 1995.

[17] R. Romano, D. Beymer, and T. Poggio. Face verification for real-time applications. In *Image Understanding Workshop*, San Francisco, 1996.

[18] M. Swain and D. Ballard. Color indexing. *Int. Journal of Computer Vision*, 7 (1), 1991.

[19] D. Swets and J. Weng. Discriminant analysis and eigenspace partition tree for face and object recognition from views. In *International Conference on Automatic Face and Gesture Recognition*, Killington, VT, October 1996.

[20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[21] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *International Conference on Automatic Face and Gesture Recognition*, Killington, VT, October 1996.

Figure 7. Identification of Richard despite one false hit. Note the pose and scale variations.

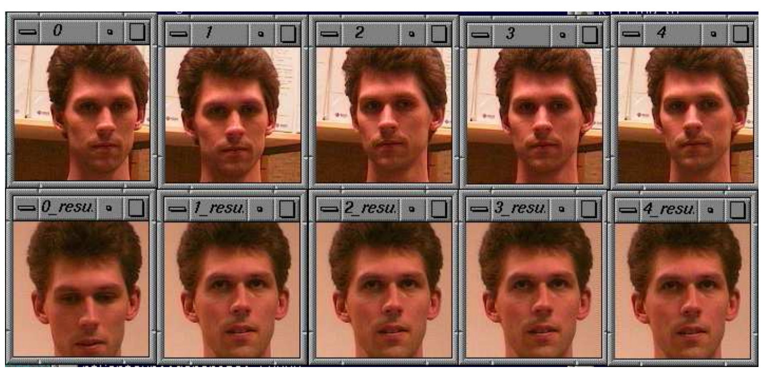

Figure 8. Identification of Matthias. Note the scale variation.