FACE RECOGNITION USING DCT–BASED FEATURE VECTORS

Christine Podilchuk and Xiaoyu Zhang

Signal Processing Research Department
AT&T Bell Laboratories
600 Mountain Avenue, Murray Hill, NJ 07974
chrisp@research.att.com

ABSTRACT

Face recognition has many applications ranging from security access to video indexing by content. We will describe an automatic face recognition system which is VQ–based and examine the effects of feature selection, feature dimensionality and codebook size on recognition performance in the VQ framework. In particular, we examine DCT–based feature vectors in such a system. DCT–based feature vectors have the additional appeal that the recognition can be performed directly on the bitstream of compressed images which are DCT–based. The system described here consists of three parts: a preprocessing step to segment the face, the feature selection process and the classification. Recognition rates for a database of 500 images shows promising results.

1. INTRODUCTION

In recent years, automatic face recognition has become a popular area of research. An excellent survey paper on the topic appeared recently in [1]. Recognition, verification and identification of faces from still images or video data have a wide range of commercial applications including video indexing of large databases, security access and other multimedia applications.

In this paper, we attempt to look at face recognition from a traditional signal processing point–of–view and adapt some of the ideas from speech recognition [2] to image recognition. A recognition system as shown in Figure 1a consists of three parts: a preprocessing step to segment the data and extract critical areas or features, feature selection and classification. The preprocessing step for face recognition could be a rough segmentation to isolate the face data from the background data or a more detailed segmentation in order to accurately locate facial parts such as the eyes, nose and mouth which will be used to generate the features. Except for applications where the data is collected in a very controlled way, preprocessing is necessary to normalize the data or extract feature vectors such as the geometrical relationships between the facial parts. Much of the work on locating faces and facial parts is based on matching a deformable template to an edge map of the input image [1]. Edge detectors are based on local operations which are very sensitive to initial conditions and noise. We introduce a new technique for detecting the critical areas of the face which is based on matching the image to a map of invariant facial attributes associated with specific areas of the face. This technique is very robust because it relies on global operations over a whole region of the face. The initial face recognition algorithm described here is VQ–based with a minimum distance classifier. A codebook of feature vectors or codewords is determined for each person from the training set. Within the VQ–framework, we examine recognition performance based on feature selection, number of features or codebook size, and feature dimensionality. For feature selection, we have examined several block–based transformations and the k–means clustering algorithm to generate the codewords for each codebook. We show that block–based DCT coefficients produce good low–dimensional feature vectors with high recognition performance. This offers the possibility of performing recognition directly on a DCT–based compressed bitstream without having to decode the image. This is especially attractive since current image compression standards both for still images and video, are DCT–based.

2. FACIAL SEGMENTATION

We introduce a framework for locating the facial features that is robust to varying conditions in lighting, posture, scale and position. This facial feature extraction algorithm could be useful as a front–end for a face recognition system either to normalize the data or pro-

X. Zhang was a University Relations student at Bell Laboratories, summer 1995

0-7803-3192-3/96 $5.00©1996 IEEE

vide critical features for classification. In our current recognition system, the segmentation algorithm is used only to normalize the data and remove extraneous areas such as the background. The algorithm is based on a general template which outlines different regions of the face. The template provides a generic outline of certain facial areas such as the forehead, eyes, cheeks, nose and mouth. The template is matched to a particular image location where a set of *a priori* constraints associated with the different areas are best met. The constraints are chosen to be invariant over a wide set of facial characteristics and external conditions such as lighting. A critical difference between this technique and many of the other face detection algorithms, is that this technique is based on global attributes associated with the particular regions outlined by the template while many other techniques rely on local operations such as edge detection.

The original idea of using a template of facial invariants was introduced by Sinha in [3]. In the original work, Sinha proposed using ratios of intensity averages over certain areas of the face. We found that the constraints proposed by Sinha define a large set of feasible solutions and could result in false positives. We modified the original ratio template idea by adding more *a priori* constraints to reduce the set of possible solutions. The set of constraints consist of facial invariants associated with the regions outlined by the template. We have considered the following attributes for the face:

- Chrominance constraints - skin tones occupy a certain range of color space. The template areas representing the cheeks should fall within the acceptable range of skin tones.

- Frequency constraints - a simple smoothness measure as given by the variance of a particular region; for example, the cheek regions are expected to be smoother than the eye regions.

The match between the template and input image is found by maximizing the cost function:

$$C = \sum_{i=1}^{N} w_i f(a(i,T), a(i,I)) \qquad (1)$$

where N is the number of attributes, $a(i,T)$ is the expected value of attribute i in the template T, $a(i,I)$ is the measured value of the attribute i from the input image I, w_i is the weight corresponding to the confidence level of attribute i and f(*) is a function of the attributes. In order to deal with image scale, the maximum of the cost function is determined by searching all spatial locations at every scale.

The idea of using a template–based set of invariants to locate facial areas could be extended to other objects. The template shape, attributes and weights could be learned over a large training set. This could be especially useful for video indexing by content. Figure 1b and 1c illustrate some of the results of the template–based technique where a modified version of the original template is shown highlighted in pink on the images. The templates shown here for illustrative purposes, highlight the eyes, cheeks, mouth and forehead. Note that these areas do not necessarily represent the critical areas for face recognition. They are being outlined to illustrate the results of the segmentation algorithm. The images shown here exhibit varying conditions in terms of lighting, scale, facial expression and position. The images in Figure 1b are professional photos from a Corel database which might be typically found in a large multimedia database. These images may have unusual lighting or background conditions. The images in Figure 1c are more typical of the type of images that may be found in a security application.

3. FEATURE SELECTION AND CLASSIFICATION

Several techniques for face recognition have been proposed which are based on extracting critical facial parts or the geometrical relationship between facial parts as the identifying features to be matched. These methods depend on the ability to locate the facial parts with very high accuracy which can become very difficult if the images are not acquired in a very controlled environment. Other techniques which do not depend on locating facial parts for the feature selection process include [4] on using the Karhunen–Loeve transform for face recognition and [5] on "eigenfaces" for face detection and recognition. Published work on the "eigenface" approach show very good results on a large database (7562 images of $\approx$ 3000 subjects) [6]. In [7], the authors compare some algorithms where a template–based scheme yields superior results to a feature–based scheme. These algorithms and others are described in the survey paper [1].

The framework used here for face recognition is based on generating a vector quantizer (VQ) codebook of feature vectors or codewords for each person in the database. The signal–processing approach taken here is an extension of some of the ideas applied to speech recognition [2]. The k–means algorithm, applied to blocks of data or transformed blocks of data from the original image, clusters the data into feature vectors. A separate codebook of features is found for each subject in the database. The statistical clustering procedure

finds the features for classification automatically. By clustering the data into features using the k–means algorithm and applying a *minimum distance* classifier, we focus on the effects of feature selection, feature dimension and codebook size on the recognition performance. For feature selection, we compare several block–based transforms with block–based pixel intensities. Figures 1d and 1e illustrate some of our findings. The recognition rates are based on a small database of 500 images representing 25 people. The results shown here are based on a training set of four images and a testing set of 16 images for each person. Sample images from the database are illustrated in Figure 1f and include variations in position, expression, scale, and with and without glasses. When this database was collected, no attempts were made to vary the lighting or background. In particular, we are interested in seeing how block–based DCT coefficients perform as feature vectors in such a scheme due to their compatibility with current compression standards. Figure 1d shows recognition rates for different codebook sizes, block sizes and feature dimension for a DCT–based scheme. The horizontal axis represents the codebook size and for each case, codebooks of size 16, 32, 64 and 128 codewords were generated. The vertical axis represents the recognition performance of the algorithms on the 500 still image database. The curves represent different input block size and feature dimension for DCT–based feature vectors. The results illustrated in Figure 1d were obtained without the segmentation algorithm. Because the background is constant and uniform for all the images in the database, good recognition results were obtained without segmentation as shown in Figure 1d. Note that larger block size and codebook size result in better performance. However, at the larger block size of 32, increasing the feature dimension from 8 to 16 does not produce a statistically significant improvement in performance. We found that by using DCT–based feature vectors, we are able to greatly reduce the feature dimension in comparison with using pixel intensity–based feature vectors without degrading the recognition performance. For example, DCT–based feature vectors of dimension 16 yield the same recognition rate as pixel–based vectors of dimension 256 – 94%. Figure 1e illustrates the improvement in using the template–based segmentation algorithm described earlier to preprocess the data. This example illustrates how the performance of the DCT–based feature vectors of block size 16 and feature dimension 8 greatly improves with segmentation. Since this database contains a constant, uniform background, the improved performance of the recognition algorithm with segmentation illustrated in Figure 1e is mostly due to the normalization of the data after segmentation to compensate for scale differences. For a database with various backgrounds, the segmentation algorithm has the potential of providing greater gains than illustrated in this example. The initial recognition results are encouraging. Further studies on larger databases are being investigated as well as using spatial constraints in the recognition algorithm. We are also investigating extending the segmentation algorithm to a more general video indexing application.

4. REFERENCES

[1] Rama Chellappa, Charles L. Wilson and Saad Sirohey, "Human and Machine Recognition of Faces: A Survey", *Proceedings of the IEEE*, May 1995.

[2] L.R. Rabiner and B-H Juang, *Fundamentals of Speech Recognition*, Prentice–Hall, New Jersey, 1993.

[3] Pawan Sinha, "Learning and Using Qualitative Invariants for Recognition", *MIT A.I. Memo No. 1505*, Nov. 94.

[4] L. Sirovich and M. Kirby, "Low–dimensional procedure for the characterization of human face", *J. Opt. Soc. Amer.*, vol 4, pp. 519–524, 1987.

[5] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces", *Proc. Int. Conf. on Patt. Recog.*, pp. 586–591, 1991.

[6] A. Pentland, B. Moghhaddam, T. Starner, and M. Turk, "Viewbased and modular eigenspaces for face recognition", *Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog.*, pp. 84–91, 1994.

[7] Roberto Brunelli and Tomaso Poggio, "Face Recognition: Features versus Templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Oct. 1993.

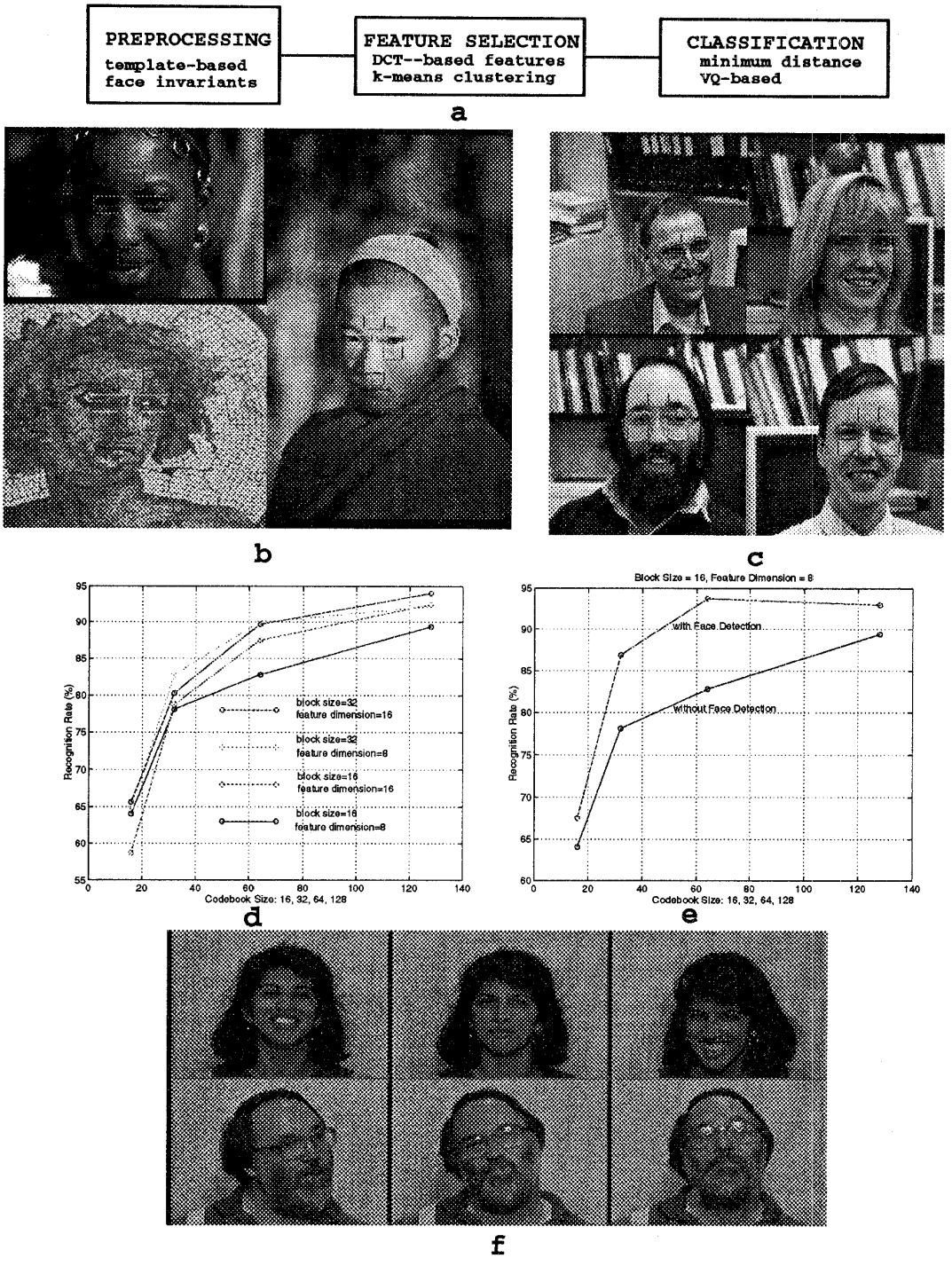

Figure 1: 1a) recognition block diagram 1b,c) segmentation results 1d,e) recognition rate vs codebook size 1f) sample images