



ELSEVIER

Signal Processing: *Image Communication* 00 (2000) 000–000

SIGNAL PROCESSING:

IMAGE
 COMMUNICATION

www.elsevier.nl/locate/image

A reliable descriptor for face objects in visual content

 Wenyi Zhao^a, Dinkar Bhat^{b,*}, N. Nandhakumar^b, R. Chellappa^a
^a*Center for Automation Research, The University of Maryland, College Park, MD, USA*
^b*LGERCA, 40 Washington Road, Princeton Jn, NJ 08550, USA*

Abstract

We present a descriptor for human face objects in visual content. The descriptor enables similarity-based retrieval using a face image as the query. The descriptor for a set of face objects consists of three components: a face subspace that is computed using principal component analysis, a discriminant matrix that classifies the set of faces, and a collection of face vectors with each vector corresponding to a particular face object. Each face vector is computed by projecting the face image onto the face subspace and then onto classification space using the discriminant matrix. In the classification space, faces of a person are distinctly clustered, and hence it becomes simpler to classify a novel image when projected onto that space. Similarity is measured in terms of the Euclidean distance measure. We demonstrate the efficacy of the descriptor for similarity-based retrieval using MPEG-7 test content. We also discuss how the descriptor satisfies some key requirements of MPEG-7. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Face descriptor; Face recognition; Biometrics; Image retrieval; MPEG-7

1. Introduction

“*I never forget faces, but in your case I will be glad to make an exception*”, said Groucho Marx. Presumably, Mr. Marx would have been delighted with a tool that searches for faces (except possibly one) in his photo-album.

Searching and retrieving parts of videos or still images containing human faces is a very useful capability that could be supported by image or video database software. A wide range of users including video producers, security personnel, and home video enthusiasts would benefit from automatic face image retrieval. The database could be a collection of home videos, a digital album of photographs, a production video library, mug shots of criminals, etc. Then, there are applications where the searchable database is not static. In a broadcast scenario with digital video content, one could match thumbnail images of a favorite actor within frames of the video stream, with the intent that scene clips of the actor be captured. Fig. 1 depicts the process of matching face descriptors for retrieval. In all these applications, the ability to use query face images to retrieve multimedia data that contain faces similar to the query is required. The key issue is choosing an

*Corresponding author. Tel: + 609-716-3515; + 609-716-3503.

E-mail addresses: wyzhao@cfar.umd.edu (W. Zhao), dbhat@lgerca.com (D. Bhat), nandhn@lgerca.com (N. Nandhakumar), rama@cfar.umd.edu (R. Chellappa).

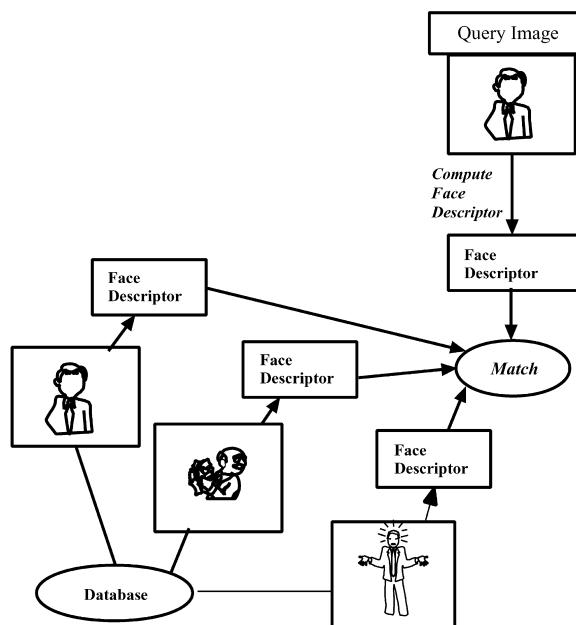


Fig. 1. A sketch illustrating a retrieval system based on face descriptors.

appropriate *face descriptor* that can be associated with multimedia data in the database. The face descriptor, in MPEG-7 terminology, describes the syntax and semantics of a representation entity for faces. While several representations are possible, developing a universal scheme for multimedia has clear benefits. It enables matching faces across databases without having to re-compute compatible descriptors. For instance, faces in a home video management system from one vendor can be matched directly with faces in a different home video system. Additionally, standardization helps in developing software libraries and applications based on face recognition that can be used with different databases. For instance, a PC user authentication application that uses face recognition can be ported across platforms with different face databases. MPEG-7 seeks to develop such a standardized face representation and resulting descriptor. In this paper, we describe a face descriptor that was proposed in the MPEG-7 evaluation meeting at Lancaster, UK, in February this year. The descriptor was evaluated favorably towards becoming part of the MPEG-7 experimentation model that is being developed.

The aim of the face descriptor is it should succinctly represent a face image, and should be able to distinguish or *classify* one face image from another. The set of faces to be distinguished is divided into classes, with images of any one person belonging to one class. Each face image occupies a point in a very high-dimensional space, the number of dimensions corresponding to the size of an image. Computation of the descriptor for the set of faces involves two steps. First, from a large “universal” set of faces that is a superset of the set of faces to be classified, a face subspace or *eigenspace* is computed using principal component analysis [15]. The selected principal components of the distribution of faces, or the set of largest eigenvectors of the covariance matrix of the set of face images, represent the most significant variation among the “universal” set of face images. This step results in a low-dimensional representation – a subspace – of the original space. Second, each face image from the set of faces to be classified is projected on to the face subspace, from which a linear discriminant matrix is computed that optimally separates face classes. When points in the face subspace (those resulting from projection of face images onto the face subspace) are

projected onto classification space using the discriminant matrix, they are clustered into distinct face classes. Any new face image can be classified by projection onto face subspace and then onto classification space.

The problem of face representation for subsequent face recognition has been addressed in considerable depth in computer vision literature. The approaches developed are far too numerous to be listed here; for in-depth survey, the reader is referred to [1]. Here, we discuss some methods that are very relevant to our approach [14,12,15]. In [11,7] the use of principal component analysis for face recognition was explored in much detail. Like in our approach, a low-dimensional representation of the original space was used, but the second stage of classification using discriminant analysis was not used. Instead, the average of the projections of faces onto the subspace, of each person, was used as the class representative vector from which distance was measured for similarity. Our approach of discriminant analysis results in optimal separation of face classes, which results in better face recognition [8]. In [4], linear discriminant was used directly on face images for classification. It was shown in [14] that the performance was poorer when compared to the cascaded approach of principal component and discriminant analysis. Parametric eigenspaces for general object recognition was developed in [4]. Samples of each object class in the low-dimensional subspace were interpolated to obtain a continuous manifold. Euclidean distance from this manifold was used as the similarity criterion for object recognition. In [10], the use of principal component analysis and linear discriminant analysis for image retrieval and general object recognition was explored. In [15], the use of PCA and linear discriminant analysis for face recognition was demonstrated by exploiting the characteristics of face objects. It was also shown that recognition rate does not degrade when the face size used varies, a practically very useful implication. The techniques mentioned above, categorized as *appearance-based methods*, show significant promise.

A good face descriptor must possess several properties. It must be robust in the sense that the descriptor does not vary widely with changing illumination, or variation in scale and orientation of the face. The descriptor must be sufficiently detailed for successful recognition, yet not so fine that would make it too sensitive to relative changes in face orientation, illumination, etc. – the classic tradeoff between false positives and false negatives. Further, to be practically useful, the descriptor must be computationally tractable, it must be extensible, it should be hierarchically expressible to allow for scalability in applications. Towards this end, MPEG-7 has established a set of requirements that must an ideal descriptor must possess. We discuss how the descriptor meets a subset of those requirements.

2. Details of the descriptor

In this section, we present a formal definition of our descriptor for face recognition, and motivation for the approach used. We envision that our descriptor would be part of a larger face Description Scheme (DS) which would enable indexing to the appropriate face image. The description scheme would include syntax like bounding box information and duration for which the face lasts in a video, and semantic information contained in the face like the depicted expression. It could also include information about groups of faces in an image or video frame like spatial relationships.

2.1. Definition

Consider a set of M face objects $F = \{f_1, f_2, \dots, f_M\}$ stored in a database, where f_k is an image vector of dimension $P \times 1$. f_k is constructed by arranging pixels of a two-dimensional face image in a sequential, raster-scan fashion, $f_k = \{f_{k,1}, f_{k,2}, \dots, f_{k,P}\}^T$. The descriptor associated with each face is denoted by $c_k = \{c_{k,1}, c_{k,2}, \dots, c_{k,N}\}$. The descriptor for the database includes a set of N basis vectors $\Phi = \{\phi_1, \phi_2, \dots, \phi_N\}$ that spans a subspace of the face vectors with $N \ll P$. The dimension of each basis vector is equal to the dimension of the original face vector. The set Φ is static (normative and computed only

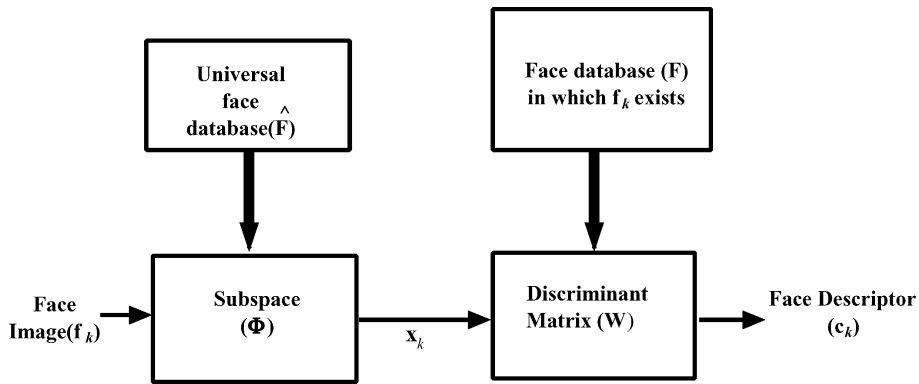


Fig. 2. Diagram that illustrates computation of the descriptor for a face image.

once) and is common to all databases. The descriptor associated with each human face object c_k is obtained by projecting the face object f_k onto Φ , and then transforming the resulting vector by a matrix W . The matrix W is called the *discriminant matrix*. This matrix could be specific to each database that allows querying and retrieval of face objects. The transformation of f_k to c_k is denoted by $f_k \xrightarrow{\Phi} x_k \xrightarrow{W} c_k$, where x_k is an intermediary representation of dimension $N \times 1$. Hence, the *descriptor D* for the set of face objects F is given by

$$D = \{C, \Phi, W\} \tag{1}$$

where the set of face images is described by $C = \{c_1, c_2, \dots, c_M\}$. Φ and W are included in the descriptor because subsequent matching requires that the query image be projected in appropriate subspaces. Fig. 2 shows the process of computing the descriptor.

As noted earlier, for retrieval, there must be a mechanism for associating C with F . For instance, a description scheme might be formulated as D, I, p , where p represents a vector of indices to specific regions/portions of still images or video frames in the database corresponding to F , and I indicates a vector of priority values denoting the relative weights to the indices. Note that the descriptor can be stored locally or remotely. In the case of remote access, additional appropriate fields may be required to be stored in the description scheme, for instance, the location of the database itself.

2.2. Motivation

Images in a face image set of different people tend to be correlated since they all represent a common visual object. This correlation denotes redundancy in data. An obvious step would be to take advantage of the redundancy in the data, and compress the data into a low-dimensional representation that preserves only significant variations between data samples. This low-dimensional representation results from projecting each face image onto the face subspace Φ . The eigenvectors constituting the subspace can be thought of as a set of features that account for the most variance within the set of faces. As mentioned earlier, the subspace is obtained using principal component analysis (PCA).

While principal component analysis reduces dimensionality of data, it does not explicitly deal with separating data into *classes*. In our case, each face is a class and images of that face form sample data for that class. By separating face images into classes, one can classify or label a novel face image. Class separation is achieved by discriminant analysis of data, which in our case is the result of face images projected onto Φ .

Discriminant analysis produces a discriminant matrix W that defines the basis vectors of a space in which data is separated optimally. We concentrate on *linear* discriminant analysis (LDA) where the surface of separation between any two classes is a hyper-plane.

But why should discriminant analysis be performed on face images that have been projected onto Φ , rather than on input face image data directly? As noted earlier, projection of a face image onto the subspace Φ preserves features that are significantly different from other face images. Subsequently, it is easier to classify faces using those features rather than the entire face data. In other words, faces in Φ that are then projected onto W are better clustered and classified than when raw image data is used. This observation was verified experimentally in [14] where the two approaches were compared.

3. Extracting the descriptor

In this section, we discuss how the components of the descriptor can be extracted. The extraction of the descriptor, in MPEG-7 usage, forms the non-normative part. Extraction strategies are left to the innovation, only the syntax of the descriptor is to be standardized.

Φ is computed from a “universal” collection \hat{F} of face image objects as follows. Each face image is first converted to a vector f_j (of dimension $P \times 1$) by arranging all the intensity values of pixels in the face image in a sequential raster-scan fashion. The average face vector h is computed for the “universal” set F . The matrix R is constructed after subtracting h from each face vector:

$$R = [f_1 - h, f_2 - h, \dots, f_M - h].$$

Subtracting the average vector ensures that the eigenvector with the largest eigenvalue represents maximum variance in the image set. The dimension of R is $P \times \hat{M}$. To compute the eigenvectors of the image set, the covariance matrix is constructed as:

$$Q \equiv RR^T$$

The dimension of Q is $P \times P$. The eigenvectors ϕ_i and corresponding eigenvalues λ_i of Q are determined by solving the eigenvalue problem:

$$\lambda_i \phi_i = Q \phi_i. \quad (2)$$

Although there are P eigenvectors for Q , only the eigenvectors corresponding to the largest N eigenvalues are retained to form Φ .

There are two issues involved: (a) How does one choose N ? (b) What is an efficient algorithm for computing Φ ? The second issue is important but not crucial because the subspace is not expected to be computed for every database, in other words, it is expected to be a normative component of an MPEG-7 compatible database which is computed only once. One way of choosing N would be to use the cumulative eigenvalue curve $c(x)$ defined as:

$$c(x) = \sum_{i=1}^x \lambda_i \bigg/ \sum_{i=1}^P \lambda_i$$

where $\lambda_i, i = 1, \dots, P$, are eigenvalues that have been arranged in monotonically decreasing order. $c(x)$ is a monotonically increasing function that denotes the amount of information retained in the subspace if $x < P$ eigenvectors are used to span the subspace. By using an appropriate threshold for $c(x)$, top N eigenvalues and corresponding eigenvectors, can be retained.

Computing the subspace using Q can be computationally expensive, the dimension of Q being $P \times P$ with P being large in general. But one can take advantage of the peculiarity of our problem. When the number of

samples \hat{M} is much smaller than P , we can compute the subspace of $\mathbf{Q}' = \mathbf{R}^T \mathbf{R}$ which is of dimension $\hat{M} \times \hat{M}$. It turns out that the eigenvectors of \mathbf{Q} can be obtained from those of \mathbf{Q}' by simply pre-multiplying them by \mathbf{R} (the eigenvalues are the same). Thus, the subspace is computed much more efficiently. There are other computationally effective ways too; for that the reader is referred to [10].

\mathbf{W} is computed for each database that contains face objects, using linear discriminant analysis. First, each face image vector \mathbf{f}_k is transformed to an intermediate representation \mathbf{x}_k by projecting the former on to the eigenspace defined by Φ :

$$\mathbf{x}_k = [\phi_1, \phi_2, \dots, \phi_N]^T (\mathbf{f}_k - \mathbf{h})$$

where \mathbf{h} is the average vector of all face vectors. \mathbf{W} is constructed to be the optimal linear discriminant function (\mathbf{x}_k) that maps a vector \mathbf{x}_k onto a classification space that best distinguishes between different classes of faces. The mapping is given by $g(\mathbf{x}_k) = \mathbf{W}^T \mathbf{x}_k$. \mathbf{W} is computed from a set of training vectors by solving the generalized eigenvalue problem:

$$\mathbf{S}_b \mathbf{W} = \mathbf{S}_w \mathbf{W} \Lambda \tag{3}$$

where \mathbf{S}_b is the *Between-Class Scatter Matrix*, \mathbf{S}_w is the *Within-Class Scatter Matrix*, and Λ is the diagonal matrix of eigenvalues. \mathbf{S}_b and \mathbf{S}_w are computed as follows:

$$\mathbf{S}_b = \frac{1}{R} \sum_{i=1}^R (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

$$\mathbf{S}_w = \frac{1}{M} \sum_{k=1}^M (\mathbf{x}_k - \mathbf{m}_{i(k)})(\mathbf{x}_k - \mathbf{m}_{i(k)})^T$$

where R denotes the number of classes that the faces can be grouped into (each class may represent an individual or a family of individuals), M denotes the total number of face images used in training, \mathbf{m}_i denotes the mean of sample vectors in the i th class, and \mathbf{m} denotes the mean of all face vectors used for training, and $\mathbf{m}_{i(k)}$ denotes the mean vector of the class “ i ” to which \mathbf{x}_k belongs.

Solving the eigenvalue problem is equivalent to maximizing the between-class scatter while minimizing the within-class scatter, i.e. maximizing the ratio of the determinants of $\mathbf{W}^T \mathbf{S}_b \mathbf{W}$ and $\mathbf{W}^T \mathbf{S}_w \mathbf{W}$. This achieves best class separation in the linear sense. The dimension of \mathbf{W} is $N \times N$ when the number of classes in \mathbf{F} is greater than N . If the number of classes is less than or equal to N , say U , then the dimension of \mathbf{W} is $N \times (U - 1)$. There are several ways to solve this generalized eigenvalue problem. One way is to directly compute the inverse of \mathbf{S}_w and then solve a non-symmetric eigenvalue problem for matrix $(\mathbf{S}_w)^{-1} \mathbf{S}_b$, which may be not numerically stable. Instead we recover a symmetric eigenvalue problem by decomposing matrix \mathbf{S}_w [15].

The last step is the computation of the face descriptors in the database. For that, each face vector in \mathbf{F} is projected on to the eigenspace defined by Φ and then onto the classification space using \mathbf{W} to form the descriptor \mathbf{c}_k for that face image object:

$$\mathbf{x}_k = [\phi_1, \phi_2, \dots, \phi_N]^T (\mathbf{f}_k - \mathbf{h})$$

$$\mathbf{c}_k = \mathbf{W}^T \mathbf{x}_k$$

The dimension of each face descriptor in \mathbf{C} depends on the dimension of \mathbf{W} which, as noted earlier, depends on the number of face classes in \mathbf{F} . If the number of classes is greater than or equal to N , then the dimension of \mathbf{c}_k is $N \times 1$. If the number of classes U is less than or equal to N , then the dimension of \mathbf{c}_k is $(U - 1) \times 1$.

3.1. An example

Below, the process of computing the face descriptor for a particular database F is described. There were 10 classes in the database representing people of different races. There are approximately 5 images per class.

Φ is computed using a “universal” database of more than 1200 segmented faces. This database contains male and female faces of different races and colors, and it is a superset of F . The background between the set of faces is held approximately constant. Each face image is normalized such that the distance between the pupils of the eyes is fixed. The line joining the eyes is horizontal. The aim of normalization is to obtain a set of faces with fixed locations of the eyes in the image coordinate system. Each segmented face region was re-scaled to a size of 48×42 pixels. In the experiments, this step is performed manually. Hence, the dimension of each face vector is 2016×1 , i.e., $P = 2016$. The subspace of the face space was computed using principal component analysis as explained in the previous section, and only 300 eigenvectors which correspond to the 300 highest eigenvalues were retained to form Φ , i.e., $N = 300$.

Unlike the computation of Φ that requires a large database of faces representing the universe of faces, W only requires F . Each face image in F is normalized as described earlier, before the between and within scatter matrices are computed, and the eigenvalue problem is solved. Since the number of classes ($= 10$) is smaller than N , W is of size 300×9 . Finally, C is obtained by projecting each face vector in F onto Φ , and then onto W , i.e., $f_k \xrightarrow{\Phi} x_k \xrightarrow{W} c_k$. c_k is a vector of dimension 9×1 .

4. Similarity measure for retrieval

To support similarity-based retrieval, distance measures are defined that measure similarity between a query face and database face objects, using their descriptors. The computed distances can then be sorted to produce a ranked list of database face objects that are similar to the query face object.

To find the set of face objects of the database F that are similar to a query face vector f_q , the following steps may be followed. The input query face image vector is scaled to dimension $P \times 1$, the canonical size used in computing the database face descriptors. The resulting input vector is projected onto each eigenvector ϕ_i to obtain x_q :

$$x_q = [\phi_1, \phi_2, \dots, \phi_N]^T (f_q - h),$$

where h is the average of the “universal” face object database that was used in constructing Φ . x_q is then projected onto the classification space using W to obtain c_q . To compare with the query face object with each of the database face objects, we suggest the Euclidean distance or the weighted Euclidean distance as candidates for d_{qk} . The Euclidean distance measure is defined by:

$$d_{qk} = \sqrt{\sum_{i=1}^N (c_{q,i} - c_{k,i})^2}. \quad (4)$$

A weighted Euclidean distance measure may be defined by:

$$d_{qk} = \sqrt{\sum_{i=1}^N \alpha_i (c_{q,i} - c_{k,i})^2}$$

where α_i are weights (based on eigenvalues) obtained during training. The N lowest distances correspond to the N most similar face objects in F . In addition, the measure can be used to rank retrieved faces.

5. Experiments

The performance of the face descriptor for retrieval of face image objects in a database was evaluated using MPEG-7 Test Content Set S4. This set contains a total of 178 face images obtained from 14 different people (classes). Of the 178 images, 140 views are frontal views, and the remaining 38 are non-frontal (profile) views of faces.

The set Φ is computed from a universal collection \hat{F} containing more than a thousand face image objects. Φ contains 300 eigenvectors, each of size 2016×1 . The following two test cases were considered. In the first case, both frontal and non-frontal face images are used in constructing F , but in the second case only frontal images are used. Thus, the queries in first case can be frontal or non-frontal. The aim is to test the performance of the linear discriminant analysis for classification under the two conditions.

5.1. Test cases

(a) From each of the 14 classes, 5 images were selected (a total of 70 images) for computing the discriminant matrix W . One non-frontal view in each set of 5 images was selected (Fig. 5a). These 70 images were stored in the database F . Each one of the 178 images available was used as a query image, and the retrieval from the database was performed using the similarity measure described above. The set of retrieved images, for each query image, were ordered based on the similarity measure. Using the criterion that the *top ranked* retrieved image must correspond to the correct class, an overall correct retrieval rate of 86.5% was observed. Using the criterion that *one of the best three* retrieved images must belong to the correct class, the correct retrieval rate of 90.4% was observed. The receiver operator curve for this test case is shown in Fig. 3.

(b) For each of the 14 classes, 4 images were selected for computing the discriminant matrix W . All 4 images were frontal views unlike case (a). The resulting 56 images were stored in the database. Each of the 140 frontal view images available was used as a query image, and the retrieval from the database was performed using the method described above. The images retrieved were ordered based on the similarity measure. Using the criterion that the *top ranked* retrieved image must correspond to the correct class, an overall correct retrieval

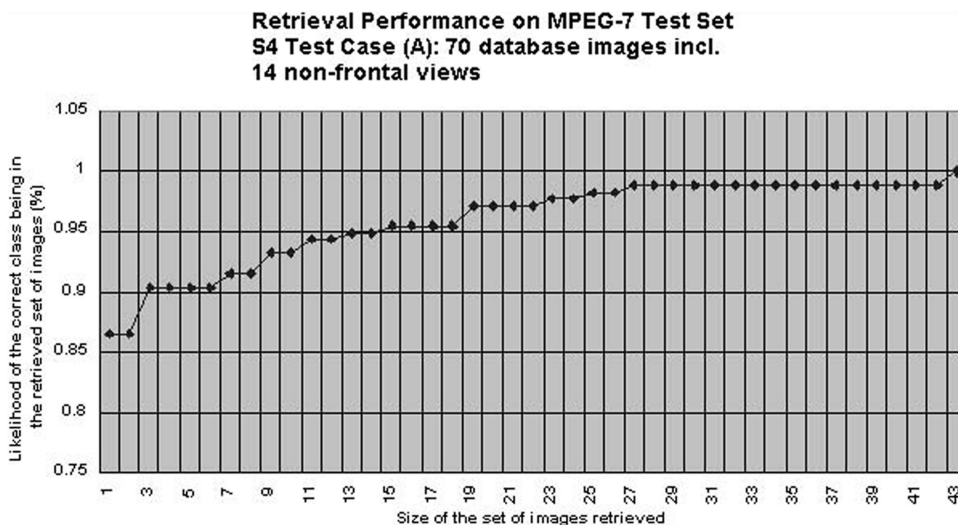


Fig. 3. Likelihood that the correct class was retrieved in the top N retrieved images for each query, for various values of N. For $N > 42$, likelihood is 100%.

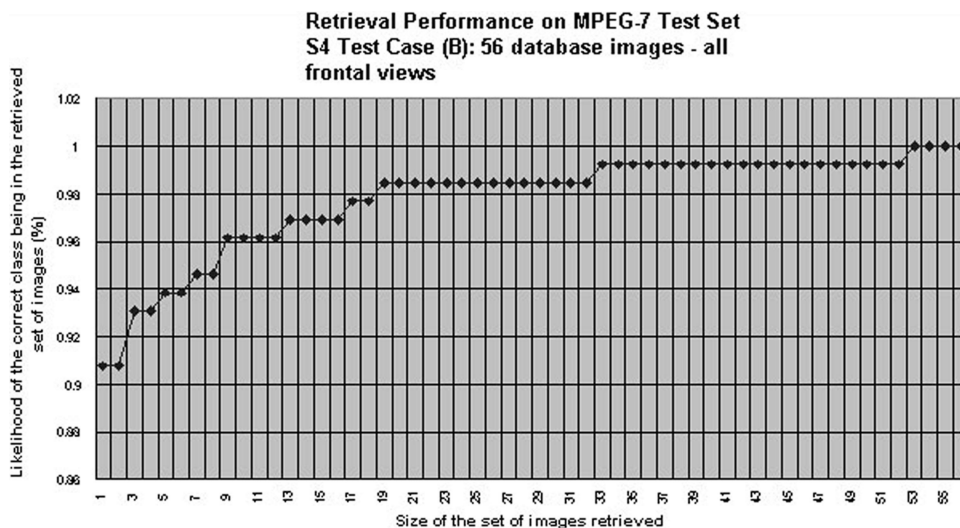


Fig. 4. Likelihood that the correct class was retrieved in the top N retrieved images for each query, for various values of N. For $N > 52$, likelihood is 100%.

rate of 90.7% was observed. Using the criterion that *one of the best three* retrieved images must belong to the correct class, the correct retrieval rate of 93.1% was observed. The receiver operator curve for this test case is shown in Fig. 4.

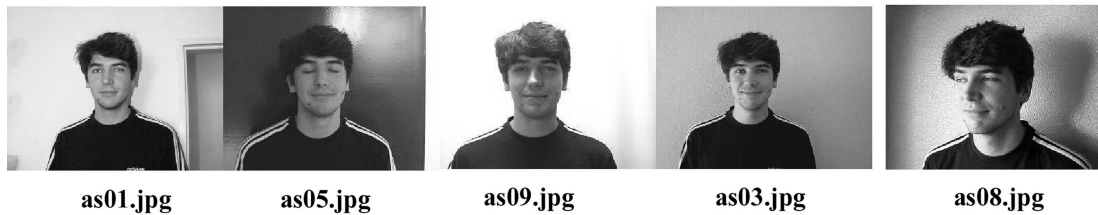
As expected, the performance of the descriptor in test case (b) is superior to that in case (a).

5.2. Example results

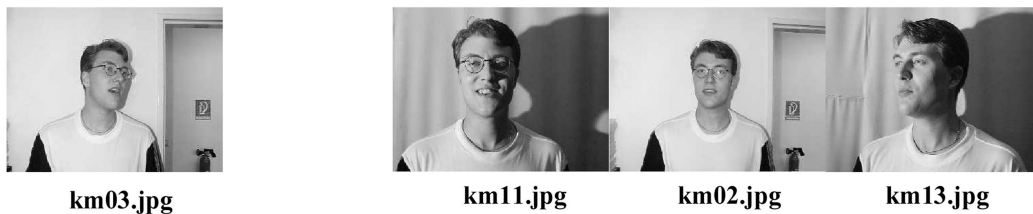
Three examples of querying are shown in Fig. 5, with the database constructed as in test case (a). In each case, the top 3 retrieved images are shown. Recall that in test case (a), the database contains both frontal and non-frontal images, and the query image could be frontal or non-frontal. In the following examples, the query images are different from those in the database (training and test sets are different).

The intention of test case 1 was to show that recognition performance with non-frontal faces as query improves when non-frontal images are added to training, as opposed to using frontal images only (we have not shown the results with non-frontal images as queries with purely frontal image training). However, we do not claim that the approach solves the problem of non-frontal image recognition, in fact, it is best suited for frontal face image recognition.

Another issue is the problem of segmenting faces and normalizing them to a canonical size before being input to the recognition algorithm. In the above test cases the segmentation was done manually, but in an operational system a face detection component would feed segmented faces to recognition component after normalization. In our prototype system, we use a training-based detection algorithm for face detection [6]. Important face characteristics are learnt from a large database of faces after they have been projected onto a set of wavelet bases. We use wavelet bases because prominent face characteristics are clearly separated from those less prominent. Subsequently, the learnt characteristics are used to classify faces from non-faces using a classifier. Unlike the recognition problem where there are as many face classes as there are number of people to be classified, the detection problem deals with two classes – face and non-face.

Training Images from One Class

(a)

Example 1**Example 2****Example 3**

(b)

Fig. 5. Results with test case a) where the training database contains both frontal and non-frontal images: (a) Training images from one class, amongst others, that are used for computing the discriminant matrix. (b) Retrieval results with three different queries. The name of the image file in the MPEG-7 test set is provided.

6. Conformance with MPEG-7 requirements

MPEG-7 has prescribed several requirements that an ideal descriptor must meet. In our proposal [5], we discuss properties of the descriptor against each requirement specifically. Here, we discuss some key requirements and how our descriptor meets them.

Regarding range of applicability, the descriptor can be associated with variety of media including still images, collections of images, and video. It may be used at different levels of abstraction in a description scheme, e.g., it can be associated with specific face regions in an image at the lowest level of abstraction, or associated with a high-level description of a collection of people, say faculty in a school. Further, multiple descriptions can be associated with a given face, for instance, different values of c_k , computed using different discriminant matrices can be associated the face. For instance, two discriminant matrices can be used, one obtained using frontal views only, and another obtained with both frontal and non-frontal views. Thus, retrieval can be attempted with different descriptions.

An important requirement for a descriptor is with respect to scalability: does the descriptor scale in proportion to size of data? The descriptor scales in two ways: (a) when the input image resolution changes, the size of each eigenvector changes in direct proportion, (b) when the number of classes in the database is varied, the dimension of the discriminant matrix varies in direct proportion (unless the number of classes is greater than the number of eigenvectors). These observations in-turn imply that the descriptor can be stored hierarchically, a coarse representation corresponding to low resolution face images with few classes, and a fine representation corresponding to high resolution face images with finer division within face classes. Consequently, similarity can be computed in a hierarchical fashion.

As discussed earlier, the proposed descriptor supports similarity-based retrieval. Using the retrieval scheme, interactive queries like “Find all faces similar (or dissimilar) to the query image” can directly be supported. With some additional information, perhaps stored in the description scheme, queries like “Find faces to the left of faces that are similar to the query face” can be handled. Thus, the descriptor is useful to applications like user authentication, browsing of home videos/security videos, etc. that are within the scope of MPEG-7 applications. As far as coding efficiency is concerned, the proposed descriptor requires only a small number values to represent each face object in the database. An additional small overhead is incurred to store the eigenvectors – this is constant for all databases – and a reasonably sized discriminant matrix.

The descriptor has been evaluated against several other schemes in the FERET test [8] that was conducted by an independent group of researchers in the US Army Research Labs. The results of comparison with other schemes are reported in [9]. The study found that a descriptor, consisting of a combination of PCA and LDA schemes, significantly outperformed various other schemes. Thus, the efficacy of the descriptor is favorable.

6.1. Integration into the description definition language

The syntax for the description definition language (DDL) is being developed using XML Schema as the basis [3]. Some color, texture and motion descriptors and associated description schemes have been described in that syntax. The following is a tentative syntax for the face descriptor based on the current DDL syntax:

```
< DescType name = 'FaceRecogDescriptor' / >
  < attrDecl name = 'W_matrix' >
    < datatypeRef name = 'matrix' / >
  < /attrDecl >
  < attrDecl name = 'Phi_matrix' >
    < datatypeRef name = 'matrix' / >
  < /attrDecl >
```

```

    < attrDecl name = 'c_vector' >
      < datatypeRef name = 'vector' / >
    < /attrDecl >
  < /DescType >

  < datatype name = 'matrix' >
    < sequence minOccur = 'm' maxOccur = 'm' >
      < sequence minOccur = 'n' maxOccur = 'n' >
        < datatypeRef name = 'real' / >
      < /sequence >
    < /sequence >
  < /datatype >

  < datatype name = 'vector' >
    < sequence minOccur = 'n' maxOccur = 'n' >
      < datatypeRef name = 'real' / >
    < /sequence >
  < /datatype >

```

Note that the syntax is subject to change, and has been described here for illustration purposes. As mentioned earlier, the face recognition descriptor would most likely be part of a larger description scheme.

7. Conclusion

We presented a descriptor for face objects in visual content that could be part of an MPEG-7 compatible database. We described extraction of the descriptor that forms the non-normative part of MPEG-7. By controlled experiments, we showed that the descriptor is effective in terms of retrieval using MPEG-7 test content. We also discussed how the descriptor satisfies some key MPEG-7 requirements.

Future work will concentrate on improving the performance of retrieval using better preprocessing techniques like illumination compensation, developing algorithms for incrementally updating the discriminant matrix when new faces are added to the database, and sensitivity of the descriptor to changing face pose. Currently, to compensate for illumination variation, a face mask and a heuristic approach based on face symmetry are applied [12]. Recently, some systematic approaches have been suggested for dealing with illumination variation for face recognition (for example, [13,2]) which may be useful for us too.

Several important issues have yet to be dealt with, for instance, coding of the descriptor in MPEG-7 streams. First, how should the linear discriminant matrix be transmitted? Clearly, it is inefficient to transmit the matrix with every face descriptor. It may be more efficient to transmit the matrices as a separate stream, and allow the descriptor for each face to point to the appropriate discriminant matrix. Second, given that the subspace matrix is normative, where should it be stored? Should it be stored with each MPEG-7 compatible database, or should there be a global location for the matrix?

Standardization of the description scheme that holds the face descriptor is key to useful applications. The description scheme must be flexible enough to contain a set of descriptors for face recognition given that MPEG-7 would likely standardize a set, rather than any single descriptor. If the description scheme is rich enough to contain details about the location of features on the face like the eye, mouth, etc., then applications like facial expression recognition can be enabled. In summary, we believe that the face description is a vital aspect of MPEG-7, with potentially wide applications.

Acknowledgements

We thank Jia Wang for her help in implementation of prototype face recognition software.

References

- [1] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces, a survey, *Proc. IEEE* 83 (1995) 705–740.
- [2] A.S. Georghiades, D.J. Kriegman, P.N. Belhumeur, Illumination cones for recognition under variable lighting: faces, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998, pp. 52–58.
- [3] J. Hunter, Proposal for a new MPEG-7 description definition language proposal, *MPEG-7 Meeting*, Vancouver, 1999.
- [4] H. Murase, S.K. Nayar, Visual learning and recognition of 3D objects from appearance in structured environments, *Int. J. Comput. Vision* 14 (1) (1995) 5–24.
- [5] N. Nandhakumar, D. Bhat, W. Zhao, R. Chellappa, Proposal 551, *MPEG-7 adhoc meeting*, Lancaster, February 1999.
- [6] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, Pedestrian detection using wavelet templates, *The Proceedings of Computer Vision and Pattern Recognition (CVPR97)*, Puerto Rico, June 1997.
- [7] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1995, pp. 786–793.
- [8] P.J. Philips, H. Moon, P. Rauss, S.A. Rizvi, The FERET evaluation methodology for face recognition algorithms, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 137–143.
- [9] S.A. Rizvi, P.J. Phillips, H. Moon, A verification protocol and statistical performance analysis for face recognition algorithms, *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1998, pp. 833–838.
- [10] D.L. Swets, J. Weng, SHOSLIF-O: SHOSLIF for object recognition and image retrieval – phase II, *Technical Report CPS-95-39*, 1995.
- [11] M. Turk, A. Pentland, Eigenfaces for recognition, *J. Cognitive Neurosci.* 3 (1991) 72–86.
- [12] W. Zhao, Improving the robustness of subspace FR system, *Proceedings of the Second International Conference Audio and Video-based Person Authentication*, Washington, DC, 1999, pp. 78–83.
- [13] W. Zhao, R. Chellappa, Robust face recognition using symmetric shape-from-shading, *Technical Report CAR-TR-919*, University of Maryland, Center for Automation Research, 1999.
- [14] W. Zhao, R. Chellappa, A. Krishnaswamy, Discriminant analysis of principal components for face recognition, *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Japan, 1998, pp. 336–341.
- [15] W. Zhao, R. Chellappa, P.J. Phillips, Subspace linear discriminant analysis for face recognition, *Technical Report CAR-TR-914*, University of Maryland, Center for Automation Research, 1999.



Wenyi Zhao received Bachelor degree in Electronic Engineering from TsingHua University, Beijing, China, and M.S. degree in Electrical Engineering from University of Virginia, in 1990 and 1995, respectively. He is expecting his Ph.D. degree in Electrical Engineering from University of Maryland, College Park by the end of 1999.

Currently he is a Graduate Research Assistant at Center for Automation Research, University of Maryland, where he is pursuing research in the areas of image analysis and understanding, pattern recognition, multimedia applications, and particularly robust face recognition. During 1997–1998, he visited LG Electronics Research Center of America, Inc., where he conducted collaborative research with other members of technical staff in the areas of video indexing and retrieval, development of MPEG-7 standard. From 1994 to 1995, he conducted research in the Machine Vision Lab, University of Virginia, where his research focused on wide-baseline stereo vision system. From 1990 to 1993, he was an engineer with Beijing Huahuan Elec. Corp. Ltd., where he led a project of image processing system design and was involved with development of digital communications systems. Then he worked at Department of Electronic Engineering, TsingHua University as a technical staff, designing application systems. Mr. Zhao is a member of Eta Kappa Nu and IEEE.



Dinkar Bhat received his B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Madras, M.S. in computer science from the University of Iowa with a thesis in computer animation, and his Ph.D. from Columbia University with a thesis in the area of Computer Vision. He is currently a member of technical staff at LG Electronics Research Center of America, conducting research and development in the area of digital television. He has been involved in different research areas including stereo vision, face detection/recognition, MPEG-7, and video indexing. He has also been a member of a team that developed software for monitoring and generating ATSC compliant digital TV streams. His current research interest is algorithms for image and video

processing within the scope of digital television. He has published in the IEEE Transactions of Pattern Analysis and Machine Intelligence, the International Journal of Computer Vision, and several conferences.



N. Nandhakumar received the B.E. (Hons) degree in Electronics and Communication Engineering from the P.S.G. College of Technology, University of Madras, India, the M.S.E. degree in Computer, Information and Control Engineering from the University of Michigan, Ann Arbor, and the Ph.D. degree in Electrical Engineering from The University of Texas at Austin. He is currently Director of Software Products & Technology at the LG Electronics Research Center of America. He directs technology development teams that are developing final products and exploring product concepts for future interactive multimedia consumer appliances including interactive DTV systems, and internet-connected video appliances. Previously, as a senior member of the technical staff at Electroglas,

Inc, he was responsible for leading the development of core technology necessary for creating a new product line of semiconductor wafer inspection stations. He has also served as Assistant Professor of Electrical Engineering and Director of the Machine Vision Laboratory at the University of Virginia. He has published more than 80 papers in refereed journals, conference proceedings, and books; and has served on the organising committees of several international conferences and workshops in the area of video processing, image analysis and computer vision. He is a Senior Member of the IEEE.



Rama Chellappa received the B.E. (Hons.) degree from the University of Madras in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. degrees in Electrical Engineering from Purdue University in 1978 and 1981, respectively. Since 1991 he is a Professor of Electrical Engineering and an affiliate Professor of Computer Science at the University of Maryland in College Park. He is also affiliated with the Center for Automation Research (Associate Director) and the Institute for Advanced Computer Studies. Prior to joining the University of Maryland, he was an Associate Professor and Director of the Signal and Image Processing Institute at the University of Southern California. Over the last eighteen years

he has published numerous book chapters and peer-reviewed journal papers. Several of his journal papers have been reproduced in Collected Works published by IEEE Press, IEEE Computer Society Press and MIT Press. He has edited a collection of Papers on Digital Image Processing (published by IEEE Computer Society Press), co-authored a research monograph on Artificial Neural Networks for Computer Vision (with Y.T. Zhou) published by Springer Verlag, and co-edited a book on Markov Random Fields (with A.K. Jain) published by Academic Press. He has served as an associate editor for IEEE Transactions on Signal Processing, Pattern Analysis and Machine Intelligence, Image Processing, Neural Networks, and as a co-Editor-in-Chief of Graphical Models and Image Processing, published by the Academic Press. He has received several awards, including the 1985 NSF Presidential Young Investigator Award, a 1985 IBM Faculty Development Award, the 1991 Excellence in Teaching Award from the School of Engineering at

USC, and the 1992 Best Industry Related Paper Award from the International Association of Pattern Recognition (with Q. Zheng). He has been recently elected as a Distinguished Faculty Research Fellow (1996–1998) at the University of Maryland.

He is a Fellow of the IEEE and the International Association for Pattern Recognition. He has served as a General and Technical program Chair for several IEEE International and National Conferences and Workshops. His current research interests are image compression, automatic target recognition from stationary and moving platforms, surveillance and monitoring, automatic design of vision algorithms, synthetic aperture radar image understanding, and commercial applications of image processing and understanding.