Face Recognition using SIFT Features

Mohamed Aly <malaa@caltech>

CNS186 Term Project Winter 2006

Abstract

Face recognition has many important practical applications, like surveillance and access control. It is concerned with the problem of correctly identifying face images and assigning them to persons in a database. This paper proposes using SIFT features [4] for the recognition process. The new technique is compared with well-established face recognition algorithms, namely Eigenfaces [7] and Fisherfaces [6, 2]. The results show the superiority of the new method over these two methods, specially using smaller training sets.

1 Introduction

Face recognition is essential in many applications, including mugshot matching, surveillance, access control and personal identification, and forensic and law enforcement applications [8]. In this problem, we have a database of a face images for a group of people. Then, given an unknown face image, we want to answer the question: "which person in our database does this image belong to?" Many algorithms and techniques have been proposed for solving such a problem [8]. This paper proposes a new approach based on Scale Invariant Feature Transform (SIFT) [4]. The new approach is compared with the well established algorithms of Eigenfaces [7] and Fisherfaces [6, 2]. The results show improved recognition accuracy over these two methods.

2 Background Information

2.1 Eigenfaces

Eigenfaces are based on the dimensionality reduction approach of Principal Components Analysis (PCA) [7]. The basic idea is to treat each image as a vector in a high-dimensional space. Then, PCA is applied to the set of images to produce a new reduced subspace that captures most of the variability between the input images. The Principal Component Vectors (eigenvectors of the sample covariance matrix) are called the Eigenfaces. Every input image can be represented as a linear combination of these eigenfaces by projecting the image onto the new eigenfaces space. Then, we can perform the identification process by matching in this reduced space. An input image is transformed into the eigenspace, and the nearest face is identified using a Nearest

Neighbor approach. Two versions of Nearest Neighbor classifiers can be used. The first compares the input image against all the images in the database. The second, called Nearest Cluster Center, computes the means of each cluster (faces of the same person) and uses those cluster means for comparison.

2.2 Fisherfaces

Fisherfaces [6, 2] approach is based on Fisher's famous Linear Discriminant Analysis (LDA) [3]. LDA is a supervised dimensionality reduction method that aims at finding linear combinations of the data that maximize the between class variability while minimizing the within class variability i.e. it tries to find a new reduced subspace that provides the best separation between the different classes in the input data. This basic idea is applied to face recognition in a manner similar to applying the PCA. Each face image is considered a point in a higher dimensional space. Then, LDA is applied to the data to get the new basis vectors, called the Fisherfaces. Face images are then projected onto this basis, where the matching is performed. We can again use the two flavors of Nearest Neighbors described above for the matching process.

2.3 SIFT

Scale Invariant Feature Transform (SIFT) features are features extracted from images to help in reliable matching between different views of the same object [4]. The extracted features are invariant to scale and orientation, and are highly distinctive of the image. They are extracted in four steps. The first step computes the locations of potential interest points in the image by detecting the maxima and minima of a set of Difference of Gaussian (DoG) filters applied at different scales all over the image. Then, these locations are refined by discarding points of low contrast. An orientation is then assigned to each key point based on local image features. Finally, a local feature descriptor is computed at each key point. This descriptor is based on the local image gradient, transformed according to the orientation of the key point to provide orientation invariance. Every feature is a vector of dimension 128 distinctively identifying the neighborhood around the key point.

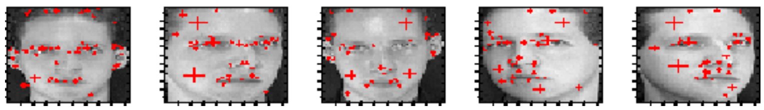
3 Approach

We propose an approach based on SIFT features for face recognition. The SIFT features are extracted from all the faces in the database. Then, given a new face image, the features extracted from that face are compared against the features from each face in the database. The face in the database with the largest number of matching points is considered the nearest face, and is used for the classification of the new face.

A feature is considered matched with another feature when the distance to that feature is less than a specific fraction of the distance to the next nearest feature. This guarantees that we reduce the number of false matches. This is because in case of a false match, there will be a number of other near features with close distances, due to the high dimensionality of the features. On the other hand, in case of a correct match,

Figure 1: AT&T Face Database

(a) 5 sample images



(b) images with SIFT features

it is unlikely to find another feature that is too close due to the highly distinctive nature of SIFT features

4 Results

4.1 Databases

Two benchmark databases are employed for comparison purposes. The first is AT&T face database [5], containing 400 images for 40 persons with 10 images/person. There are different orientations and facial expressions for each subject. The image size is 112×92 pixels. There is an average of 70 SIFT features extracted from each image. Figure 1 shows a sample of images for one subject.

The second database is Yale face database [1]. It contains 165 images for 15 subjects, with 11 images/person. The images contain different facial expressions and illumination conditions for each subject. The image size is 243×320 pixels, and an average of 230 SIFT features are extracted for each image. Figure 2 shows a sample of images from this database.

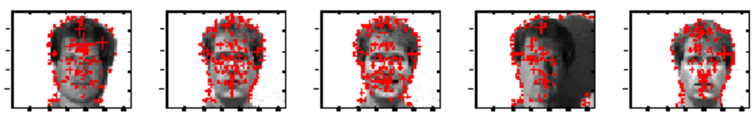
The raw faces were used without any kind of preprocessing (cropping, normalization, histogram equalization, etc.) to assess the robustness of the algorithms in the comparison.

4.2 Algorithms

Matlab was used to implement the Eigenfaces, Fisherfaces, and SIFT matching algorithms. The code for extracting SIFT features was available from David Lowe's website.

Figure 2: Yale Face Database

(a) 5 sample images for a subject



(b) Faces with SIFT features shown as crosses

4.3 Experiments

4.3.1 Baseline Comparison

The new approach was compared to both Eigenfaces and Fisherfaces. Ten independent runs were carried out, where the dataset was split randomly into two halves, one for training and one for testing. The results introduced are the average of these runs. Three distance measures were used for Eigenfaces and Fisherfaces: euclidean distance $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$, city-block distance $d(x, y) = \sum_i |x_i - y_i|$, and the cosine distance $d(x, y) = \frac{x \cdot y}{||x||_2 \, ||y||_2}$, where x & y are two feature vectors, and $|| \cdot ||_2$ denotes the euclidean norm. Both Nearest Neighbor and Nearest Cluster Center were used with the two algorithms. Two distance measures were used for matching SIFT features: cosine distance and the angle distance defined as $d(x, y) = \cos^{-1}(x \cdot y)$.

The results in table 1 clearly show the superiority of the SIFT technique over the other two methods. For AT&T, SIFT achieved 96.3% accuracy, compared to 92.9% for Eigenfaces and 93.8% for Fisherfaces. The same applies to Yale database, where SIFT got 91.7% compared to 72.1% for Eigenfaces and 86.9% for Fisherfaces. The results also show that the city-block distance is generally better for Eigenfaces and Fisherfaces, while the angle distance is better for SIFT matching.

4.3.2 Training Set Size

Two more experiments were carried out to check the performance with different training set sizes. The first was run using training set of size 20% and test set of 80%, while the second using 80% training and 20% testing. In all the experiments, 10 independent trials were performed with randomly chosen training and test sets. Table 2 shows the results. As expected, the performance degrades with smaller training set size and increases with larger training set. It is also clear that the SIFT approach is better than the others. The performance is significantly better in Yale database using the smaller training set (90.1% for SIFT vs. 73.3% for Eigenfaces and 83.5% for Fisherfaces).

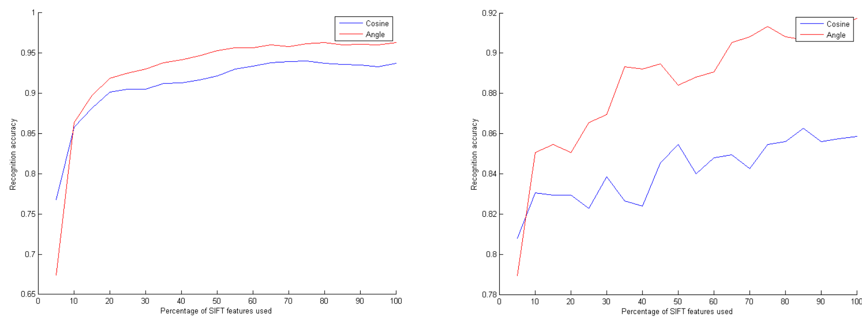
Table 1: Baseline Accuracy Results. The best results are in boldface.

	Eigenfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T	89.3	***92.9***	89.0	74.7	87.1	73.7
Yale	68.4	72.0	68.0	57.7	**72.1**	59.4
	Fisherfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T	91.3	90.8	***93.8***	91.4	91.1	93.7
Yale	83.4	86.8	86.4	83.8	**86.9**	84.6
	SIFT					
	Cosine			Angle		
AT&T	93.7			**96.3**		
Yale	85.8			**91.7**		

Table 2: Training set size results

	Eigenfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T 20%	76.0	**80.1**	76.1	71.6	79.2	70.0
Yale 20%	69.5	**73.3**	72.0	58.9	69.9	62.1
AT&T 80%	96.0	**97.2**	95.5	78.6	91.3	76.5
Yale 80%	81.3	**83.0**	81.0	70.0	78.6	76.3
	Fisherfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T 20%	76.8	74.7	84.6	79.0	77.4	**85.0**
Yale 20%	83.4	82.3	82.3	**83.5**	82.5	82.0
AT&T 80%	95.2	94.1	96.0	95.6	94.6	**96.2**
Yale 80%	87.0	89.6	89.3	87.0	**89.6**	89.3
	SIFT					
	Cosine			Angle		
AT&T 20%	79.6			**85.7**		
Yale 20%	84.7			**90.1**		
AT&T 80%	99.0			**99.7**		
Yale 80%	92.0			**95.6**		

Figure 3: Significant number of SIFT features



(a) AT&T database (b) Yale database

4.3.3 Number of SIFT Features

In an attempt to assess the significant number of SIFT features required for reliable matching of face images, several experiments were performed using only a subset of the extracted SIFT features in the matching process. The SIFT features were sorted descendingly according to their scale, and only the first p% of the average number of features were used. We tried p of 5 to 100% with increments of 5. Figure 3 shows the results for AT&T and Yale databases. These are the average results for 10 independent runs, using 50% training and 50% testing.

Clearly, the accuracy increases rapidly with increasing the number of SIFT features used and then starts to saturate. Using only 30% of the features yields accuracy better than that of Eigenfaces and Fisherfaces. This can considerably decrease the run time for SIFT matching process, as the number of matching operations is $O(n^2)$ where n is the number of features to be matched. Hence, using only 30% of the features takes only 9% of the time used to match all the points.

4.3.4 Resolution of Face Images

To check the effect of down sampling the face images on the accuracy reported for SIFT feature matching, a number of experiments were performed with different face image sizes. The original sizes were scaled down to 75%, 50% and 25% of their original sizes. Ten independent runs were performed, using random splits of the data into two halves.

Table 3 shows the results. Clearly, the average number of SIFT features extracted decreases with decreasing the resolution of the image. However, the resolutions up to 50% give comparable results to those at full scale, while at 25% the accuracy decreases significantly. In fact, the results up to 50% resolution are still better than Eigenfaces and Fisherfaces.

Table 3: Resolution results

	AT&T			Yale		
Resolution	#SIFT	Cosine	Angle	#SIFT	Cosine	Angle
100%	70	93.8	96.3	230	85.9	91.7
75%	53	92.6	95.8	155	87.2	91.4
50%	30	94.7	94.9	87	87.9	92.5
25%	10	88.4	88.4	33	79.2	82.8

5 Summary

This paper presents a new approach for face recognition, based on matching SIFT features. The new approach is compared to Eigenfaces and Fisherfaces, and proved superior to both of them in all experiments, specially with smaller training set sizes. Upon investigating the effective number of SIFT features required for reliable matching, the experiments reveal that we need only 30% of the features, which saves 91% of the time needed to match all the extracted features. In addition, the SIFT features approach continues to provide superior performance for up to 50% reduction in resolution.

References

[1] Yale face database. http://cvc.yale.edu.

[2] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 19(7):711–720, July 1996.

[3] R. A. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2004.

[5] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *2nd IEEE Workshop on Applications of Computer Vision*, 1994.

[6] Daniel L. Swets and John Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 18(8):831–836, August 1996.

[7] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Proceedings CVPR '91., IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991.

[8] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.