

# Face Image Retrieval by Using Haar Features

Bau-Cheng Shen<sup>1</sup>, Chu-Song Chen<sup>1,2</sup>, Hui-Huang Hsu<sup>3</sup>

<sup>1</sup>*Institute of Information Science, Academia Sinica, Taipei, Taiwan*

<sup>2</sup>*Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan*

<sup>3</sup>*Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan*  
{bcshen,song}@iis.sinica.edu.tw, h\_hsu@mail.tku.edu.tw

## Abstract

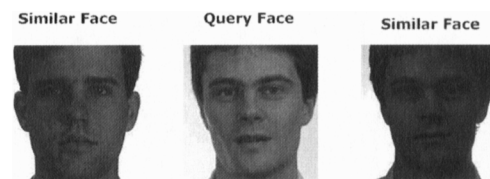
*We propose a new method to retrieve similar face images from large face databases. The proposed method extracts a set of Haar-like features, and integrates these features with supervised manifold learning. Haar-like features are intensity-based features. The values of various Haar-like features comprise our rectangle feature vector (RFV) to describe faces. Compared with several popular unsupervised dimension reduction methods, RFV is more effective in retrieving similar faces. To further improve the performance, we combine RFV and a supervised manifold learning method and obtain satisfactory retrieval results.*

## 1 Introduction

The goal of face retrieval is to retrieve face images that are similar to a specific query face image in large face databases. The retrieved face images can be used for various applications, such as photo management or visual surveillance. This retrieval task includes two kinds of target images. One is the face images with the same identity of the query face. The other is the face images which have appearance similar to the query face. Figure 1 shows search results of two kinds of target images.

Navarrete and Ruiz-Del-Solar [8] used tree-structured self-organizing map (TS-SOM) to organize face images for fast retrieval. Chow and Rahman [5] combined multiple facial features and multi-SOM for face matching. The SOM-based methods usually do not consider the face variations of the same person, but this characteristic is important in face retrieval.

Dimensionality reduction is usually desirable for analyzing data. Recently manifold learning is proposed to simultaneously reduce dimensionality and preserve local structures in the data. Manifold learning contains



**Figure 1. The center face is a query face. The left face has a different label with the query face. The right face has the same label with the query face.**

two disciplines, unsupervised [7] and supervised manifold learning. In supervised manifold learning, Local Discriminant Embedding (LDE) [3] and Marginal Fisher Analysis (MFA) [14] have shown their classification potential for face recognition. They achieve good performance because they consider the inter-face and intra-face variances by using graph embedding.

Many previous works used original face images to train a classifier or to do dimensionality reduction. However, the raw images contain pixel-wise intensity values which are vulnerable to various kinds of noises. In order to consider information in local regions and overcome noises, we transfer pixel-wise intensity values to component-wise features. There are many component-based features in the literature, such as Rectangle, HOG and EOH features [?] [6] [9].

Wu and Navetia [13] suggested that training with human annotation may not be optimal for the classification task. They used the rectangle features of one strong classifier to construct a descriptor of the target class data and use clustering to improve their classification performance. In [2], the sum of all responses of features is used to train a better face detector by using the Adaboost algorithm [?]. These works inspire us two things. One is that rectangle features can be used

to describe sub-regions of human faces, such that pixel-wise data can be transformed into component-wise features. Second, after training with the Real Adaboost algorithm, we can use all of the selected rectangle features as our feature descriptor.

In this paper we use all selected rectangle features as our rectangle feature vector (RFV) to describe face images. Supervised manifold learning is then used to construct an effective face manifold. Finally, we use the nearest neighbor method to classify new input images on the face manifold.

## 2 Manifold Learning for Face Retrieval

We select discriminant rectangle features from a large feature pool with Real Adaboost algorithm [11] and use them as our component-based features. These features are assembled as a feature vector, called the rectangle feature vector (RFV), to describe faces. Then we combine the RFV with a supervised manifold learning method, LDE, for face retrieval, and achieve promising face retrieval results.

### 2.1 Formation of the Feature Vector

To form our feature pool, we prepare a set of Haar-like features with different block sizes and types as our rectangle feature pool  $P$ . We selected a total of  $T$  features from the feature pool  $P$  by using the Real Adaboost [11] algorithm and the cascaded structure as in [4]

Given face images and non-face images as our positive and negative data. After training with cascaded Real Adaboost, we obtain  $S$  strong classifiers, where  $S$  is the number of cascaded layers. Given a face image  $x$ , the output of the  $s$ -th layer of the trained cascaded struture can be formulated as:

$$H_s(x) = \sum_{t=1}^{T_s} h_{s,t}(B_{s,t}(x)), s = (1, \dots, S)$$

where  $T_s$  is the number of weak classifiers of the  $s$ -th layer,  $B_{s,t}$  is the  $t$ -th rectangle feature value, and  $h_{s,t}$  is the classification confident value of a weak classifier which depends solely on the  $t$ -th rectangle feature value. The traditional strong classifier  $H$  uses the sum of confident values to classify this input image  $x$ . In our method, we use all rectangle features values of the  $S$  layers as our feature vector, which can be described as

$$RFV(x) = [B_{1,1}(x), B_{1,2}(x), \dots, B_{2,1}(x), \dots, B_{S,T_S}(x)]$$

### 2.2 Supervised Manifold Learning

Manifold learning approaches can represent intrinsic characteristics of high dimensional data in a lower-dimensional space. We use LDE [3] and the proposed RFV to learn a face manifold. In LDE [3], neighborhood graphs are constructed with affinity weights to formalize a graph embedding problem as an eigenvalue problem. Each input data is projected onto the low dimensional manifold, and then the nearest neighbor classifier is used to decide its class.

We show the LDE [3] algorithm below. Given data points  $\{x_i\}_{i=1}^m \in \mathbb{R}^n$  and the corresponding labels  $\{y_i \in \{1, 2, \dots, C\}\}_{i=1}^m$ . The data matrix  $X = [x_1 x_2 \dots x_m]$  is in  $\mathbb{R}^{n \times m}$ . The LDE can be realized by the follow three steps.

1. Construct neighborhood graphs.  
Let  $G$  and  $G'$  denote two different neighborhood graphs. To construct  $G$ , we consider each pair of points  $x_i$  and  $x_j$  from the same class. An edge is added between  $x_i$  and  $x_j$  if  $x_j$  is one of  $x_i$ 's  $k$ -nearest neighbors. For  $G'$ , instead of considering each pair of  $x_i$  and  $x_j$  with different classes, we connect  $x_i$  and  $x_j$  if  $x_j$  is one of  $x_i$ 's  $k'$ -nearest neighbors.
2. Compute affinity weights.  
Specify the affinity matrix  $W$  of  $G$ , where each element  $w_{ij}$  represents the weight of the edge between  $x_i$  and  $x_j$ , and is given by
$$w_{ij} = \exp(-||x_i - x_j||^2/t).$$
By default,  $w_{ij} = 0$  if  $x_i$  and  $x_j$  are not connected. The other affinity matrix  $W'$  of  $G'$  can be computed in the same way.
3. Complete the embedding.  
Find the generalized eigenvectors  $v_1, v_2, \dots, v_l$  that correspond to the  $l$  largest eigenvalues in

$$X(D' - W')X^T v = \lambda X(D - W)X^T v,$$

where  $D$  and  $D'$  are diagonal matrices with diagonal elements  $d_{ii} = \sum_j w_{ij}$  and  $d'_{ii} = \sum_j w'_{ij}$ . The embedding of  $x_i$  is accomplished by  $z_i = V^T x_i$ , where  $V = [v_1 v_2 \dots v_l]$ .

## 3 Experiments

In the first part of the experiments, we explain how to train the RFV by using Real Adaboost [11]. In the second part, we show how we prepare training data for face manifold learning and testing. At last, we analyze experimental results.

### 3.1 Feature Vector Training

We use frontal face images in CMU PIE [12]. There are 68 persons in PIE database. We rotate and resize frontal face images to crop them manually. In total we collect 6,048 face images as positive training data. The non-face data includes 17,739 images which were collected from the Internet. We randomly select 6,048 images with different sizes at different positions from these non-face images as negative data. We normalize these positive and negative data to size  $24 \times 24$ . We specify that the total detection rate is 99.7% and the total false positive rate is  $6.4 \times 10^{-3}\%$ . Then we use Real Adaboost algorithm [11] and the cascaded structure [?] to select features. We refer the readers to [11] for the detail of Real Adaboost. When the training process is completed, 232 rectangle features in six layers are selected. We use them as our RFV descriptor in the following experiments.

### 3.2 Manifold Learning and Test Stage

We use Feret [10] face databases for manifold learning. It contains 14,126 images from 1,199 individuals. We detect frontal faces by using the OpenCV face detector [1]. After detecting frontal faces, there are 3,470 frontal face images of 1,187 individuals. Table 1 shows the distribution of numbers of detected images of each individual. For example, there are 734 persons with 2 detected face images. These faces include many variations, including lighting, beard, and eyeglasses. We align and crop these detected faces according to the eyes positions and normalize them to size  $24 \times 24$ . Some face samples are shown in Figure 2. In the training stage, there are 2,374 training images in total. In the test stage, we use the other 1,096 images as test samples. The training and test samples are not overlapped. We set that both  $k$  and  $k'$  to 1 and the variance  $t$  to 50.

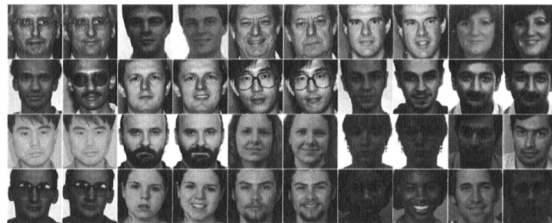
We test the performance of our RFV and face retrieval method. We use the nearest neighbor method to classify new input images. In the first experiment, we compare the ability of our RFV with unsupervised methods: row data (Ori), PCA, LPP [7]. The second experiment, we combined previous unsupervised feature with LDE: (Ori+LDE), (PCA+LDE), and (RFV+LDE).

### 3.3 Experimental Results

In first experiment, we show the error rates of different unsupervised methods with respect to different levels of preserved data energy. The preserved data energy is the preserved eigen ratio that we used for dimension reduction. In all unsupervised methods (Table 2),

**Table 1. The distribution of the number of frontal face images in Feret database.**

Images	2	3	4	5	6	7	8	9	10	11	12	13
People	734	199	136	12	43	10	28	15	2	4	2	2



**Figure 2. Faces sample images in Feret database**

the performance of our component-based feature vector (RFV) is the best when preserving 90% data energy with 232 dimensions.

In second experiment, we combined the unsupervised feature with LDE, the error rate of our method is 0.244 when preserving 90% data energy with 102 dimensions (Table 3). In combined the unsupervised feature with LDE, (RFV+LDE) and (PCA+LDE) have good performance in energy 90%. Our method (RFV+LDE) consistently outperforms other methods ( (Ori+LDE) and (PCA+LDE) ).

In Figure 3, we show the precision when retrieving the  $k$  nearest neighbors. We check 10-100 nearest neighbors to test different methods. We find our method (RFV+LDE) has better performance than (PCA+LDE). In Figure 4, we show some five retrieved similar images, along with the query image. The experimental results show that our method can effectively retrieve similar face images from a large database.

## 4 Conclusions

We have proposed a method for retrieving similar face images in large face databases. Our method uses a set of rectangle features as the RFV descriptor and exploits this descriptor with supervised manifold learning. Experimental results show that our method is effective and outperforms the other methods. In future work, we will consider relevance feedback to refine retrieval results.

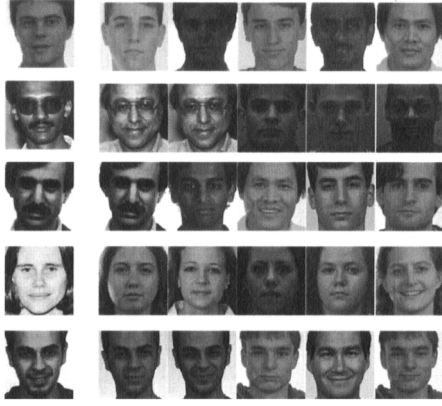


Figure 3. The query and retrieved face images

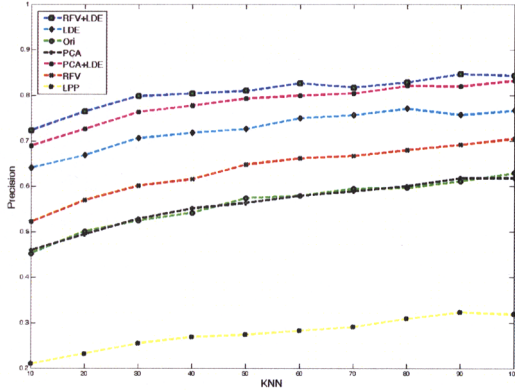


Figure 4. Precision-Knn results of different methods on Feret face database [10]

Table 2. Error rates of different unsupervised methods at different levels of preserved energy. The number in parenthesis is the preserved data dimensionality.

	E:0.90	E:0.93	E:0.96	E:0.99
Ori	0.47(576)	0.47(576)	0.47(576)	0.47(576)
PCA	0.50(61)	0.45(94)	0.49(156)	0.48(317)
LPP	0.769(404)	0.760(437)	0.748(477)	0.736(536)
RFV	<b>0.440(232)</b>	0.440(232)	0.440(232)	0.440(232)

Table 3. Error rates of different unsupervised features which combined with LDE at different levels of preserved energy. The number in parenthesis is the preserved data dimensionality.

	E:0.90	E:0.93	E:0.96	E:0.99
Ori+LDE	0.354(308)	0.359(359)	0.357(423)	0.362(522)
PCA+LDE	0.297(43)	0.292(120)	0.310(133)	0.315(148)
RFV+LDE	<b>0.244(102)</b>	0.252(111)	0.247(123)	0.281(136)

## Acknowledgement

This research was supported by the National Science Council of Taiwan under Grant No. NSC 96-3113-H-001-011.

## References

- [1] <http://www.intel.com/technology/computing/opencv/>.
- [2] L. Bourdev and J. Brandt. Robust object detection via soft cascade. *CVPR 2005*.
- [3] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. *CVPR 2005*.
- [4] Y.-T. Chen and C.-S. Chen. Fast human detection using a novel boosted cascading structure with meta stages. *Image Processing, IEEE Transactions on*, 17(8):1452–1464, Aug. 2008.
- [5] T. W. Chow and M. K. Rahman. Face matching in large database by self-organizing maps. *Neural Process. Lett.*, 2006.
- [6] N. Dalai and B. Triggs. Histograms of oriented gradients for human detection. *CVPR 2005*.
- [7] X. He and P. Niyogi. Locality preserving projections. *NIPS 2003*.
- [8] P. Koikkalainen. Progress with the tree-structured self-organizing map. *ECAI 1994*.
- [9] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. *CVPR 2004*.
- [10] P. Phillips, H. Moon, P. Rauss, and S. Rizvi. The feret evaluation methodology for face-recognition algorithms. *CVPR 1997*.
- [11] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Computational Learning Theory*.
- [12] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. *FG 2002*.
- [13] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. *ICCV 2007*.
- [14] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang. Graph embedding: a general framework for dimensionality reduction. *CVPR 2005*.