Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 105 (2007) 99–111
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Multi-view face and eye detection using discriminant features ☆

Peng Wang [a,*], Qiang Ji [b]

[a] Section of Biomedical Image Analysis, Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, USA
[b] Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Received 5 July 2005; accepted 30 August 2006
Available online 3 November 2006

Abstract

Multi-view face detection plays an important role in many applications. This paper presents a statistical learning method to extract features and construct classifiers for multi-view face detection. Specifically, a recursive nonparametric discriminant analysis (RNDA) method is presented. The RNDA relaxes Gaussian assumptions of Fisher discriminant analysis (FDA), and it can handle more general class distributions. RNDA also improves the traditional nonparametric discriminant analysis (NDA) by alleviating its computational complexity. The resulting RNDA features provide better accuracy than the commonly used Haar features in detecting objects of complex shapes. Histograms of extracted features are learned to represent class distributions and to construct probabilistic classifiers. RNDA features are subsequently learned and combined with AdaBoost to form a multi-view face detector. The method is applied to both multi-view face and eye detection, and experimental results demonstrate improved performance over existing methods.

Keywords: Multi-view face detection; Feature extraction; Recursive nonparametric discriminant analysis; Eye detection; AdaBoost

1. Introduction

Faces in images have various poses. In this paper, a face pose is defined as a set of similar face orientations, which is caused by out-of-plane head rotation. Fig. 1 shows some typical mean faces of different poses. Multi-view face processing has recently received more attentions, since more than 75% faces in real images are non-frontal [15]. Multi-view face detection plays important roles in many applications, such as surveillance, human computer interaction and entertainment.

During the last several years, the frontal face detection has obtained promising progress with the use of Haar features and the AdaBoost algorithm [34]. For the purpose of detecting both frontal and profile faces, a multi-view face detector is needed since faces of different poses show quite different appearance. For example, two eyes are the most distinguishing features for frontal face detection [34,2], while there are no such clear features in profile faces. Some other facial features, such as noses and mouth, are also blurred in profile faces. When including more pose variation in training, the resulting multi-view face detector will have high complexity to compensate the absence of clear features, therefore its generalization capability and computational efficiency will decrease in detection.

Currently most often used features for face detection are selected from a large set of local geometric features, such as Haar wavelets [34]. Although Haar features are easy to compute, they also suffer from limited discriminant capability. Especially at the later training stages in AdaBoost, the error rate of a classifier, which is based on a single Haar feature, is near 50%. Usually, it needs nearly 2000–5000 classifiers for frontal face detection while most of them only deal with less than 1.0% of non-face data [34,15].

In this paper, we provide a general feature extraction method for multi-view face detection. We present a recursive nonparametric discriminant analysis, based on traditional Fisher discriminant analysis and nonparametric

☆ The research described in this paper is supported in part by a Grant (N41756-03-C-4028) to Rensselaer Polytechnic Institute from the Task Support Working Group (TSWG) of the United States.
* Corresponding author.
E-mail addresses: wpeng@ieee.org (P. Wang), qij@ecse.rpi.edu (Q. Ji).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.08.008

Fig. 1. Mean faces of different poses. From left to right: full left profile, half left profile, frontal, half right profile and full right profile.

discriminant analysis, to extract features for more general distributions with moderate requirement of training samples and running time. Probabilistic classifiers are then learned from feature distributions. Multiple classifiers are combined together using AdaBoost to form a multi-view face classifier. The feature extraction and classification methods are also applied to eye detection in this paper.

The rest of this paper is organized as following. Related work is reviewed in Section 2. In Section 3, we present a recursive nonparametric discriminant analysis (RNDA) method to extract features for multi-view face and eye detection. The AdaBoost algorithm to combine multiple probabilistic classifiers using RNDA features is introduced in Section 4. Experimental results of multi-view face detection and eye detection are presented in Section 5. We conclude the paper in Section 6.

2. Related work

In this section, we review some typical face detection and eye detection methods, with the emphasis on the multi-view face detection. From the perspective of pattern recognition, all the frontal and multi-view face detection methods can be roughly categorized based on features or classifiers they use. The features used in face detection include intensity, color, texture, edge and facial components [21,9,39,43,8]. Some features show good performance at specific conditions, but have limitations for general cases. For example, skin color can be an effective feature to segment face regions. However, it needs heuristic postprocessing to extract faces from the segmentation results. Also, skin color is sensitive to illumination changes. Another type of features, wavelets, may handle moderate illumination variation [29,34]. When considering both accuracy and efficiency, the methods based on partial or holistic intensity usually have the best performance in different application environments [21,29,25,34]. As to classifiers, many types of classifiers have been applied in face detection, such as neural networks [25,7], Bayesian classifier [30,29], Support Vector Machine (SVM) [21], SNoW architecture [24] and AdaBoost [34].

A neural network-based face detector has been developed in [25]. In this method, normalized pixel intensity is sent to a neural network to test if the patch contains a face. The detection rate is between 74.9% and 90.3% for different tests. With a convolutional neural network architecture [7], the speed of neural network based methods is increased to 4 frames per second (fps). Only frontal face detection results are reported for this type of methods.

Support Vector Machine (SVM) has been successfully applied to frontal face detection [21], and multi-view face detection [16] due to its excellent generalization capability. In [16], a pose SVM is trained to discriminate different poses, and one SVM is trained for each typical pose. In detection, the face pose is first estimated for each testing patch with the pose SVM, then the testing patch is sent to the SVM of the corresponding pose to remove the non-face. Li et al. extract face features with the use of kernel PCA (KPCA), then apply the KPCA features in SVMs to detect faces [14]. Wang et al. combine multiple SVMs through bagging to improve detection robustness under real environments [38]. A common drawback of SVM-based methods is that their detection speed is usually slow when the number of support vectors (SVs) increases dramatically due to large pose variation.

In Naive Bayesian classifier based methods, class distributions of the face and non-face are learned from training samples, then a likelihood ratio classifier is constructed to separate the face from the non-face. In the work of [20,32], the face and non-face class distributions are modeled using mixture-of-Gaussian models, whose parameters are learned from EM [20] or K-means algorithms [32]. All these methods are only applied to the frontal face detection. Schneiderman et al. develop the first practical multi-view face detection system [30,29]. In their method, the face appearance is decomposed into some visual attributes, with each attribute represented by quantized wavelet coefficients. Assuming independence among these attributes, empirical class distributions are represented by the product of all the attributes' histograms. Based on class likelihood ratio, probabilistic detectors are constructed for both frontal and profile faces. The early implementation of their method took about a minute to detect faces in a 320 by 240 image, and its speed is increased to about 2 frames per second in the latest implementation [28]. This method needs a large number of training data to learn class distribution. It is shown in [31] that the interpolation of multi-view faces can be used to represent persons under limited face samples. Based on Schneiderman's method, Verma et al. detect faces in a video sequence, which is essentially a combination of face detection and tracking [33].

Currently, AdaBoost-based methods are popular for fast object detection since they can achieve excellent balance between detection accuracy and computational efficiency [34,41,2,15]. Viola and Jones first apply Discrete AdaBoost in frontal face detection [34]. In their method, critical features are selected from an over-complete feature

set, which may contain more than 45,000 different Haar wavelet features. Threshold classifiers are then learned from the selected features, and are combined by AdaBoost. With a cascade structure, AdaBoost-based frontal face detection methods can achieve real-time speed (i.e., above 15 frames per second) with accuracy comparable to other methods. Many efforts have been made to extend the AdaBoost method to the multi-view face detection. Some methods directly apply the same Haar features as in frontal detection for profile face detection [15,42,41], while other methods use extended Haar feature sets for profile face detection [12].

The methods directly applying the rectangle Haar features focus on improving the efficiency of AdaBoost algorithm itself. Li et al. present a FloatBoost algorithm to backtrack all the learned features and remove the redundant ones [15]. Some methods connect all the layers in a cascade together to form a nesting chain [42,41]. The chain structure allows directly utilizing training results of previous layers in the later stages, thus fewer features are needed to achieve the same performance. Some other efforts are made to extend the original rectangle Haar features to improve their limited discriminant capability for profile face and eye detection. The extended Haar sets include tilt features [17], center-surrounded features [2] and diagonal filters [12]. Current multi-view face detection can achieve a speed of 5–8 frames per second with detection rate lower than frontal face detection [15,12].

Besides the commonly used local features, it is also claimed that global features can provide better accuracy for detection. The PCA feature is applied for frontal face detection as a complementary of local Haar features [44]. However, PCA is usually used to reduce feature dimension for representing a class of objects, not for discriminating different classes. Liu et al. select features from a wavelet bank by minimizing the Kullback–Leibler divergence [18]. The method applied a 1D sequential optimization method to approximate a KL feature from a subset of local features. However, all the global features are only applied to the frontal face detection.

Current eye detection methods can be divided into two categories: active and passive eye detection [11]. The active detection methods use special illumination and IR cameras to quickly locate pupil centers. The disadvantages are that they need special lighting sources and have more false detections within an outdoor environment. Passive methods directly detect eyes from images within visual spectrum and normal illumination condition. Some early work localizes eyes based on distinct eye features, such as image gradients [13] and projection [45]. However, these features are sensitive to image noise. Huang and Wechsler select optimal Wavelet packets and classify the eye and non-eye with Radial Basis Functions (RBFs) [10]. In [3], a two-layer Gabor wavelet network (GWN) is used to localize facial points from coarse to fine. In [2,19], AdaBoost method is used to train for both face and eye detector with Haar features. After a face is detected, eyes are then located inside the face region.

3. Discriminant feature extraction for object detection

In this paper, we use pixel intensity of gray-scale images to detect faces. Although the pixel intensity is simple to use, one drawback is the high dimensionality of a face vector. For example, a face image is usually normalized to a fixed size of 20 by 20, therefore the extracted face vector has 400 elements. Based on pixel intensity, our method statistically extracts important features that can separate the face class from the non-face class.

Statistical feature extraction methods learn features from training samples without assuming any geometric shapes as Haar features do. When considering computational complexity, we only discuss linear feature extraction in this paper. Fisher discriminant analysis (FDA) provides the optimal feature for Gaussian class distributions. Nonparametric discriminant analysis (NDA) can avoid the Gaussian assumption. However, some practical problems prevent it from being directly applied in face detection. Based on FDA and NDA, we present a novel recursive nonparametric discriminant analysis (RNDA) for multi-view face detection [37]. The RNDA feature is a global feature, which is directly learned from training samples as a whole, instead of as a combination of local features [18].

In this section, we first introduce Fisher and nonparametric discriminant feature analysis, then present a recursive nonparametric discriminant analysis. Before presenting our algorithm, the notations used in this paper are introduced. A training sample is denoted as (x, g_x), where x is the data, actually a vector of image intensity in this paper. $g_x \in \{-1, +1\}$ is the sample label. The class $\Omega_1 = \{x | g_x = 1\}$ represents the object (face or eye) while the class $\Omega_2 = \{x | g_x = -1\}$ represents the non-object. Each training sample (x, g_x) is associated with a weight w_x, and $\sum_x w_x = 1$.

3.1. Fisher discriminant analysis and nonparametric discriminant analysis

3.1.1. Fisher discriminant analysis

Linear feature extraction maps original data x to a feature space y by a linear transformation F, i.e., $y = F(x) = \Phi^{\mathrm{T}} x$, where Φ is a transformation matrix or vector. A criteria to evaluate the feature y is its Bayes error E_F:

$$E_F = \int (1 - \max_i p(\Omega_i | F(x))) p(x)\, \mathrm{d}x \tag{1}$$

It has been shown that Fisher discriminant analysis (FDA) minimizes the Bayes error assuming that each class is a single Gaussian distribution, and all classes have equal priors and equal covariance matrices [40]. FDA method extracts discriminant features by maximizing the ratio of between- and within-class variance. The ratio is denoted as $J(\Phi) = \frac{\|\Phi^{\mathrm{T}} S_{\mathrm{b}} \Phi\|}{\|\Phi^{\mathrm{T}} S_{\mathrm{w}} \Phi\|}$, where S_{b} is the between-class scatter matrix, and S_{w} is the within-class scatter matrix. When each sample x is associated with a weight w_x, the scatter matrices are calculated as

$$S_w = \sum_i \sum_{x\in\Omega_i} w_x (x-\mu_i)(x-\mu_i)^T$$
$$S_b = \sum_i P(\Omega_i)(\mu_i-\mu)(\mu_i-\mu)^T \quad (2)$$

where μ is the mean of all samples, and μ_i is the mean of class Ω_i. $P(\Omega_i)$ is the weight of class Ω_i, i.e., $P(\Omega_i) = \sum_{x\in\Omega_i} w_x$. The FDA features can be extracted by solving a generalized eigenvectors problem, as in Eq. (3).

$$(S_w^{-1} S_b)\Phi_k = \lambda_k \Phi_k \quad (3)$$

The column vector Φ_k in the transformation matrix Φ is the eigenvector of $(S_w^{-1}S_b)$ corresponding to eigenvalue λ_k.

A problem of applying FDA feature to multi-view face detection is that the single Gaussian assumption in FDA is not valid for multi-view faces. For example, the frontal face needs to be modeled with a mixture-of-Gaussians [32,20]. It is even more difficult to parametrically model the non-face since the non-face is the "rest of the world" compared with the face class. The non-face class is much larger than the face class so that another assumption in FDA, equal priors for different classes, is also violated. Furthermore, for a two-class problem, the rank of S_b in FDA is 1, which means that only one effective feature can be extracted from FDA.

3.1.2. Nonparametric discriminant analysis

Nonparametric discriminant analysis (NDA) has been proposed to avoid the Gaussian assumptions in FDA and to obtain fully ranked scatter matrices from nearest neighbors (NNs) instead of from the entire class [6]. The neighborhood of a sample x is defined as $NN_x = \{x' \mid \|x'-x\| < c_x\}$, where c_x is the distance defining the neighborhood. In NDA, each sample also has extra-class and intra-class nearest neighbors, which are denoted as x^E_{NN} and x^I_{NN} respectively in Eq. (4).

$$x^E_{NN} = \{x' | g_{x'} \neq g_x, x' \in NN_x\}$$
$$x^I_{NN} = \{x' | g_{x'} = g_x, x' \in NN_x\} \quad (4)$$

In calculation, NNs are usually represented by their averaged means, i.e., $\mu^E_x = E[x^E_{NN}]$ and $\mu^I_x = E[x^I_{NN}]$. $E[\cdot]$ represents the average on the sample weights. For example, $\mu^E_x = E[x^E_{NN}] = \frac{1}{C}\sum_{x'\in x^E_{NN}} w_{x'}x'$, where $C = \sum_{x'\in x^E_{NN}} w_{x'}$

From the NNs, the nonparametric between-class scatter matrix S'_b is defined as Eq. (5).

$$S'_b = E_x[\gamma_x (x-\mu^E_x)(x-\mu^E_x)^T] \quad (5)$$

The NDA weight γ_x is defined as Eq. (6) where α is a control parameter.

$$\gamma_x = \frac{\min(\|x-\mu^E_x\|^\alpha, \|x-\mu^I_x\|^\alpha)}{\|x-\mu^E_x\|^\alpha + \|x-\mu^I_x\|^\alpha} \quad (6)$$

γ_x is close to 0.5 if a sample x is near a class boundary, and close to 0 otherwise. The NDA weight can remove the affects of the samples far away from the class boundary, so the resulting NDA features will focus on separating the samples near the class boundary. Based on NNs and nonparametric scatter matrices defined as above, the NDA feature can also be extracted by solving a generalized eigenvector problem [6,1].

Some practical difficulties prevent the nonparametric discriminant analysis from prevailing as FDA. A face vector has about 400 elements, so it needs many training samples to accurately locate nearest neighbors in such a high dimension space. Therefore, to search NNs and to compute scatter matrices in NDA is very time-consuming since there are usually more than ten thousands of face samples and many more non-face samples.

3.2. Recursive nonparametric discriminant analysis

To address the difficulties in FDA and NDA, we present a recursive scheme for nonparametric discriminant analysis. There are two innovations in the RNDA. First, NNs are searched in a transformed feature space, which has lower dimensionality than the original data space. This saves searching time and needs fewer training samples to locate NNs. Second, a recursive strategy is proposed to refine the estimation of NNs and to compute RNDA features. The algorithm is summarized in Table 1 and is explained in detail as below.

3.2.1. Searching nearest neighbors

In NDA, the nonparametric scatter matrices are calculated from NNs, so the performance of NDA depends on how accurately the nearest neighbors are located. However, as stated before, the original data space is of high dimensionality because a face vector has about 400 elements. Many training samples are needed to estimate class distributions, and the large size of training samples requires a lot of computation to locate NNs for each data. To address these difficulties, our method searches NNs in a transformed feature space instead of the original data space.

Assuming that a feature, $y = \Phi^T x$, is used to identify NNs, the neighborhood of a sample x is defined as

$$NN_x = \{x' | \Phi^T x' \in NN_{\Phi^T x}\} \quad (7)$$

Compared with Eq. (4), the neighborhood NN_x is defined in a lower dimensional space since the linear transformation Φ usually has fewer columns than rows. In our method, class distributions are represented by feature histograms, so the neighborhood of a sample x is defined as

Table 1
Recursive nonparametric discriminant algorithm

- Initialize RNDA using FDA, and obtain the initial transformation Φ_0
- Repeat the following steps until the error rate converges
  (1) Search NNs at $y = \Phi_0^T x$
  (2) Compute nonparametric scatter matrices S'_b and S'_w based on updated NNs, as Eqs. (5), (8) and (10)
  (3) Update optimal RNDA projection Φ_0 from S'_b and S'_w, as Eq. (9)
  (4) Measure the error rate $E_{F(x)=\Phi_0^T x}$, as Eq. (1)
- Output features $\Phi^T x$, where columns of Φ are the eigenvectors of $S_w^{-1}S'_b$

the bin in which $\Phi^{\mathrm{T}}x$ is located. If we set Φ as a transformation vector, NNs can be searched in an 1D feature space to reduce the required number of training samples, and $c_x = \frac{Q}{2}$ where Q is the quantization step of discriminant features. The NNs are also divided into intra- and extra-class neighbors based on their class labels, which is illustrated in Fig. 2.

Based on the identified NNs, a nonparametric within-class scatter matrix S'_{w} is defined as

$$\begin{aligned} S'_{\mathrm{w}} &= \sum_i P(\Omega_i) E_{x\in\Omega_i}\left[\gamma_x (x-\mu_x^{\mathrm{I}})(x-\mu_x^{\mathrm{I}})^{\mathrm{T}}\right] \\ &= \sum_i P(\Omega_i) \sum\nolimits_{x\in\Omega_i}\left[\frac{w_x\gamma_x}{P(\Omega_i)}(x-\mu_x^{\mathrm{I}})(x-\mu_x^{\mathrm{I}})^{\mathrm{T}}\right] \\ &= \sum_x w_x\gamma_x (x-\mu_x^{\mathrm{I}})(x-\mu_x^{\mathrm{I}})^{\mathrm{T}} \end{aligned} \tag{8}$$

where $P(\Omega_i) = \sum_{x\in\Omega_i} w_x$ is the class weight.

To separate different classes, we want to maximize the extra-class variance while minimizing the intra-class variance. Similar to FDA, the RNDA features are extracted by solving the following generalized eigenvector problem:

$$(S'^{-1}_{\mathrm{w}} S'_{\mathrm{b}})\Phi_k = \lambda_k \Phi_k \tag{9}$$

In Eq. (9), scatter matrices S'_{w} and S'_{b} are computed as Eq. (8) and (5). Since nonparametric scatter matrices S'_{w} and S'_{b} have full rank, RNDA can provide multiple features. Usually the eigenvectors Φ_0 of $S'^{-1}_{\mathrm{w}} S'_{\mathrm{b}}$ corresponding to larger eigenvalues provide better discriminant capability.

3.2.2. Recursive feature extraction

As observed from Eq. (7), an initial feature $y = \Phi^{\mathrm{T}}x$ is needed to locate NNs in one dimensional space. In the RNDA, we initialize RNDA feature with Fisher discriminant analysis which are based on parametric scatter matrices S_{w} and S_{b} as in Eq. (2). Then a recursive strategy is applied to refine the estimation of NNs and to re-compute RNDA features.

In RNDA, the NNs and scatter matrices are updated with the previous results until the estimated misclassification error converges. At each iteration in the recursive procedure, the NNs are first updated by searching at the previous RNDA feature space. Based on the updated NNs, the scatter matrices S'_{w} and S'_{b} are re-calculated. A new optimal RNDA projection Φ_0 is extracted from the new scatter matrices S'_{w} and S'_{b} by solving the generalized eigenvector problem, as in Eq. (9). The new RNDA feature $\Phi_0^{\mathrm{T}}x$ will then be used to update NNs at the next iteration. Also at each iteration, a misclassification error is estimated based on the feature $y = \Phi_0^{\mathrm{T}}x$. The iteration continues until the estimated Bayes error converges to a minimum. The whole recursive procedure is summarized in Table 1.

Furthermore, to relate the above iterations to reducing misclassification error, we define the RNDA weight γ_x in a new way, as Eq. (10):

$$\gamma_x = \frac{\min(W_x^{\mathrm{E}}, W_x^{\mathrm{I}})}{W_x^{\mathrm{E}} + W_x^{\mathrm{I}}} \tag{10}$$

where W_x^{E} and W_x^{I} are the sum of extra- and intra-class sample weights in the neighborhood, i.e., $W_x^{\mathrm{E}} = \sum_{x'\in X_{\mathrm{NN}}^{\mathrm{E}}} w_{x'}$ and $W_x^{\mathrm{I}} = \sum_{x'\in X_{\mathrm{NN}}^{\mathrm{I}}} w_{x'}$.

The RNDA weight γ_x in Eq. (10) actually represents the local misclassification error of current RNDA feature $\Phi_0^{\mathrm{T}}x$. Since w_x represents the data distribution, we have $W_x^{\mathrm{I}} = P(x' \in X_{\mathrm{NN}}^{\mathrm{I}})$ and $W_x^{\mathrm{E}} = P(x' \in X_{\mathrm{NN}}^{\mathrm{E}})$. When using the NNs to estimate data density, γ_x is actually the local misclassification error at the bin where $\Phi^{\mathrm{T}}x$ is located based on Eqs. (1) and (10), i.e., $\frac{\min(W_x^{\mathrm{E}}, W_x^{\mathrm{I}})}{W_x^{\mathrm{E}} + W_x^{\mathrm{I}}} = 1 - \max_i P(\Omega_i|\Phi^{\mathrm{T}}x)$. The total Bayes error can be approximated by weighted summation of local misclassification error, i.e., $\Gamma \approx \sum_x w_x\gamma_x$. In the recursive procedure, if RNDA converges after several iterations, which means NNs are stabilized, we have $\nabla\Gamma \approx 0$. As a result, the misclassification error also reaches its minimum. It is possible that the result corresponds to a local minimum since a global optimal solution is extremely difficult to achieve for the transformation vector with about 400 elements.

The performance of FDA and RNDA is compared using three real face data sets composed of different poses, as shown in Fig. 3. Each data set contains 5000 face sam-

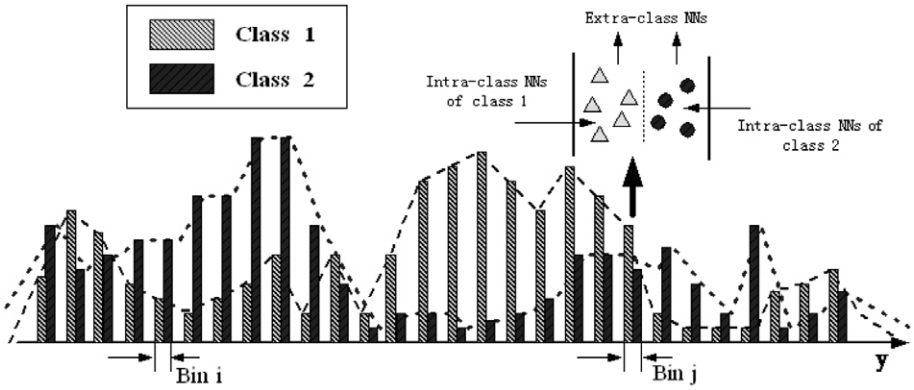

Fig. 2. Representing nearest neighbors using 1D feature histograms. Histograms of the face and non-face classes are calculated from the feature $y = \Phi^{\mathrm{T}}x$. NNs are defined as the samples whose features are located in the same bin of the histograms.

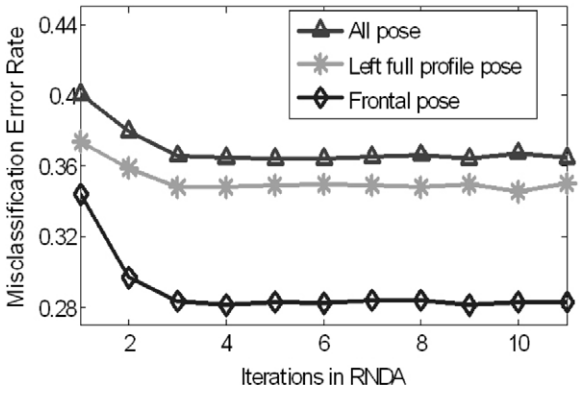

Fig. 3. Comparison of Fisher and recursive nonparametric discriminant analysis.

Table 2
Running time (in seconds) of RNDA and NDA

	Sample number			
	8000	12,000	16,000	20,000
RNDA	184.13	285.38	395.77	519.33
NDA	723.80	1456.94	2470.66	3691.11

ples and 5000 non-face samples. For each test set, the sample weights are uniformly initialized. RNDA is initialized with FDA, hence the first points in the curves represent the FDA results. For all the three data sets, only after 2 or 3 iterations, the RNDA converges with a smaller misclassification error than FDA. The RNDA is more efficient for a high dimensional space and a large training set, which is the case for face detection. An example comparing necessary running time of RNDA and NDA on a real face data set is given in Table 2. Usually, the training set contains more than ten thousand face and non-face samples, thus RNDA significantly reduces the computational complexity.

4. Constructing classifiers using RNDA features and Adaboost

Generally speaking, RNDA features can be applied to any type of classifiers. Considering both accuracy and speed, AdaBoost is used to train a face classifier. In our method, probabilistic classifiers are constructed from sequentially learned RNDA features, and then multiple probabilistic classifiers are combined together using AdaBoost to build a multi-view face detector.

4.1. AdaBoost algorithm

AdaBoost learning method is an iterative procedure to select features and combine classifiers [26]. At each iteration, it selects a feature with minimum misclassification error, and trains a weak classifiers based on the selected feature. Sample weights are updated according to current learning results, and the future learning will focus on the misclassified samples. AdaBoost learning method keeps combining new weak classifiers into a stronger one until it achieves satisfying performance.

The Real AdaBoost algorithms have been proposed in [27,5]. The weak classifiers in Real AdaBoost can output continuous confidence values, therefore has better accuracy compared with the Discrete AdaBoost [4]. Our method is based on the Real AdaBoost algorithm, which is summarized in Table 3. The weak classifier $h_t(x)$ has the form:

$$h_t(x) = \frac{1}{2}\log\left[\frac{P(\Omega_1|x)}{P(\Omega_2|x)}\right] = \frac{1}{2}\log\left[\frac{P(x|\Omega_1)}{P(x|\Omega_2)}\right] + \frac{1}{2}\log\left[\frac{P(\Omega_1)}{P(\Omega_2)}\right] \quad (11)$$

The weak classifier in Eq. (11) is actually the log ratio of posterior probabilities. It can be written as a combination of likelihood ratio and prior ratio. Usually the likelihood model is approximated by feature distributions. The prior ratio $\frac{P(\Omega_1)}{P(\Omega_2)}$ is unknown so that it needs to be determined during learning.

4.2. Applying adaboost to combine RNDA features

Both RNDA and AdaBoost algorithms utilize sample weights w_x during training. Since w_x refers to the same distribution of training samples, the RNDA feature extraction method can be naturally incorporated in AdaBoost learning algorithm. The first step of applying RNDA features in AdaBoost is to construct a probabilistic weak classifier from a learned RNDA feature. RNDA provides multiple feature candidates, $y_i = \Phi_i^{\mathrm{T}} x, i = 1, \ldots, M$, where Φ_i's are the eigenvectors corresponding to the M largest eigenvalues of $S_{\mathrm{w}}'^{-1} S_{\mathrm{b}}'$. From these eigenvectors, a vector Φ corresponding to the minimum classification error is selected. Then class distributions, $P(x|\Omega_1)$ and $P(x|\Omega_2)$, are estimated from the distribution of feature $y = \Phi^{\mathrm{T}} x$, i.e., $P(x|\Omega) \approx P(y|\Omega)$. The weak classifier in Eq. (11) can be written as

$$h_t(x) = \frac{1}{2}\log\left[\frac{P(\Omega_1|x)}{P(\Omega_2|x)}\right] \approx \frac{1}{2}\log\left[\frac{P(\Phi^{\mathrm{T}}x|\Omega_1)}{P(\Phi^{\mathrm{T}}x|\Omega_2)}\right] + \theta_t \quad (12)$$

The parameter θ_t actually represents the log ratio of priors $\frac{1}{2}\log[\frac{P(\Omega_1)}{P(\Omega_2)}]$, which is determined during training to minimize the misclassification error.

Table 3
Real AdaBoost Algorithm

- Uniformly initialize sample weights w_x
- For $t = 1, 2, \ldots, T$ (T is the maximum number of weak classifiers or when the accuracy satisfies expectation):
  (1) Select or extract a feature F_t which can minimize misclassification error, with the use of sample weights w_x
  (2) Fit a weak classifier $h_t(x)$ based on F_t
  (3) Update sample weights $w_x \leftarrow w_x \exp[-g_x h_t(x)]$ and re-normalize w_x so that $\sum_x w_x = 1$
- Output the combined classifier $H_T = \mathrm{sgn}[\sum_{t=1}^{\mathrm{T}} h_t(x)]$

In our method, the feature histogram is used to represent class distributions $P(x|\Omega_1)$ and $P(x|\Omega_2)$. The ideas are illustrated in Fig. 4, where 5000 left full profile faces and 10,000 non-faces are used in the experiment. Fig. 4(a) shows face and non-face class distributions of the first RNDA feature. The feature value is quantized into discrete bins, as in Fig. 4(b). In real training, usually there are about 10,000 face samples and more non-face samples, therefore the feature values are quantized to about 500 bins to guarantee that there are average 20 face samples in each bin.

In the next step, AdaBoost algorithm will increase weights of the samples that are wrongly classified, and new RNDA features will focus on the wrongly classified samples. This is illustrated in Fig. 4(c), where the samples wrongly classified by the first weak classifier are located inside or near region A. Specifically, face samples in the regions B and C have increased weights after the first iteration in AdaBoost. Due to re-weighting, the face class does not hold single Gaussian distribution anymore. As shown in Fig. 4(d), the face class distribution of the second feature is more like a mixture of Gaussian distributions, where the samples in region B and C form 2 clusters.

As the above iteration continues, AdaBoost keeps updating sample weights and selecting new RNDA features to construct more weak classifiers. The iteration will continue until the combined classifier achieves satisfying accuracy. The algorithm of applying RNDA in AdaBoost is summarized in Table 4.

To compare the performance of RNDA features with Haar and FDA features, a classifier is trained with AdaBoost using real profile face data. The Haar features used here include both original and tilt sets [17]. Two sets are used in this experiment, one set from half left faces and another from full left profile faces. Each set contains 5000 faces and 5000 non-faces. In the experiment, the detection rate of the face class is fixed at 99.0%, so the false positive rates of using different features are compared. The algorithm continues until the false positive rate is lower than 2.0%. The results are shown in Fig. 5. In both sets, the false positive rates decrease for all the three features. As observed from the curves, Haar features, however, have limited discriminant capability. After the first several iterations, the false positive rates only drop slightly, and end at 33.5% for half profile faces and at 40.7% for full profile faces with 50 iterations, requiring many more features to

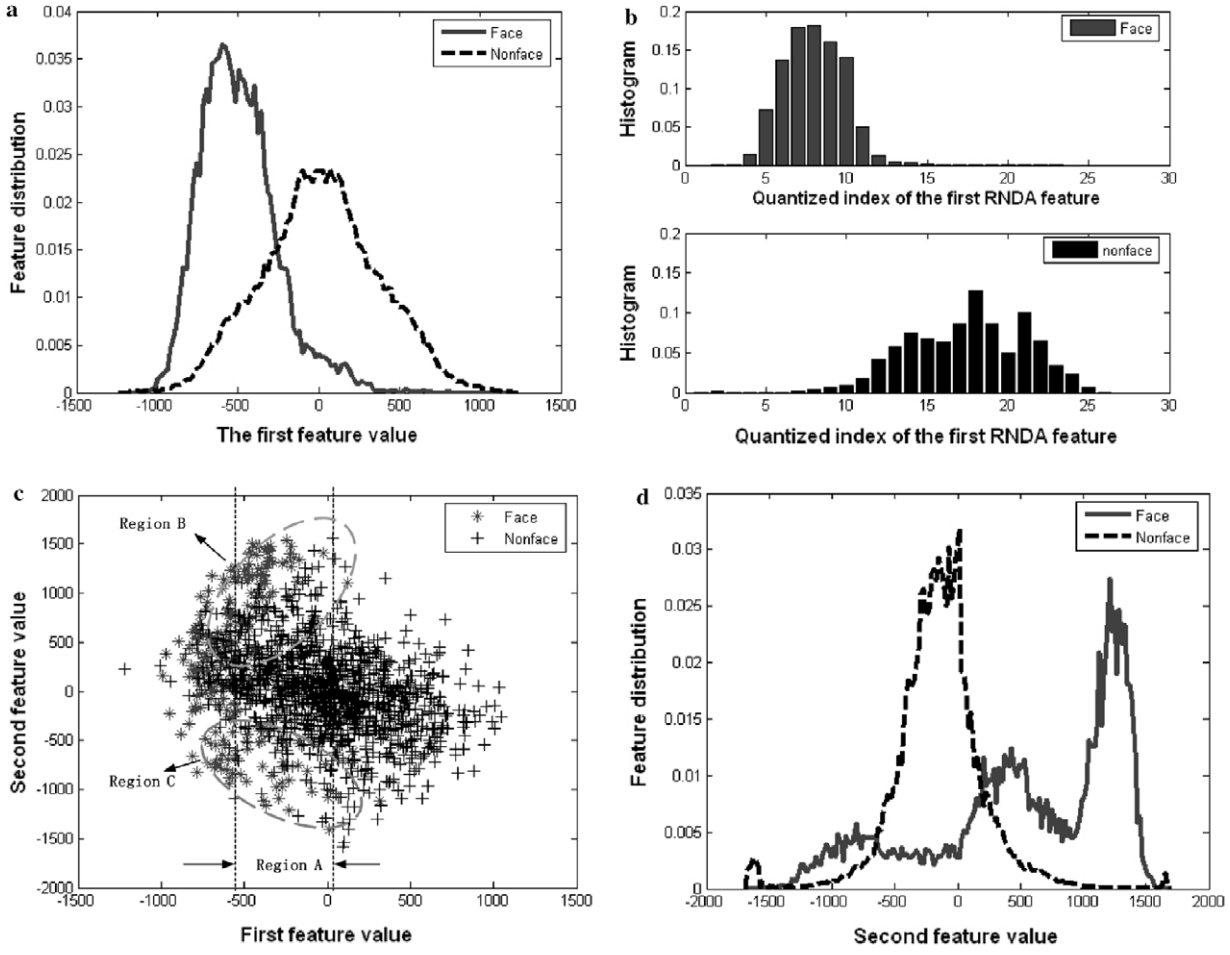

Fig. 4. Feature distribution on left full profile face data and non-face data. (a) Distribution of the first RNDA feature. (b) Histograms of the first RNDA feature. (c) Joint distribution of the first two RNDA features. (d) Distribution of the second RNDA feature.

Table 4
Applying discriminant features in Real AdaBoost

- A sample is denoted as (x, g_x), where x is the training data, and $g_x \in \{-1, 1\}$ is its label. Sample weights w_x are uniformly initialized at the beginning
- Set M as the maximum number of weak classifiers. Set an expectation of classification accuracy rate R, $t = 0$, and $R_0 = 0$
- Repeat while $t < M$ and $R_t < R$
  (1) $t \leftarrow t + 1$
  (2) Extract RNDA feature $y = \Phi^{\mathrm{T}} x$ using sample weights w_x. Approximate class distributions with the feature y, i.e., $P(x|\Omega_1) \approx P(y|\Omega_1)$ and $P(x|\Omega_2) \approx P(y|\Omega_2)$
  (3) Learn a weak classifier $h_t(x)$:
  $h_t(x) = \frac{1}{2}\log[\frac{P(\Omega_1|x)}{P(\Omega_2|x)}] = \frac{1}{2}\log[\frac{P(x|\Omega_1)}{P(x|\Omega_2)}] + \theta_t$
  where θ_t is a threshold minimizing the misclassification error
  (4) Set $w_x \leftarrow w_x \exp[-g_x h_t(x)]$ and re-normalize sample weights so that $\sum_x w_x = 1$
  (5) Obtain the current combined classifier $H_t = \mathrm{sgn}[\sum_{i=1}^{t} h_i(x)]$ and estimate classification accuracy R_t
- Output the classifier $H(x) = \mathrm{sgn}[\sum_t h_t(x)]$

remove the remaining non-faces. In contrast, Fisher and RNDA features have better discriminant capability than Haar features. For half profile faces, FDA with 45 iterations achieves a 3.5% false positive rate while RNDA features can achieve a 3.0% false positive rate with only 22 iterations. For full profile faces, with 24 iterations, FDA features achieve a 14.3% false positive rate while RNDA features can achieve a 2.5% false positive rate. It is clear from the above experiments that Fisher features are more powerful than Haar features, and that RNDA features further improve accuracy of FDA features.

4.3. Feature pool and greedy selection

The difference between FDA and RNDA is that RNDA can provide multiple features, which form a feature pool. At tth iteration in AdaBoost, the feature pool is denoted as $F_{\mathrm{pool}} = \{\Phi_1^{(t)}, \ldots, \Phi_M^{(t)}\}$. Usually the feature pool is compact, $M = 5$, while the commonly used Haar feature sets contain above 45,000 over-complete features. A feature F is selected out of the feature pool to construct a weak classifier. Traditional AdaBoost forwardly selects the feature that can minimize misclassification error based on current data weights. A greedy method is applied here to minimize the overall misclassification error instead of current misclassification error.

The overall error of combining jth weak classifier candidate, $h_t^j = \{F_t^j, \theta_t^j\}$, is defined as $E(F_t^j, \theta_t^j) = \frac{1}{2}\sum_x |g_x - sgn(H_t^j(x))| w_x$, where $H_t^j = H_{t-1} + h_t^j$. The feature and parameters are learned as

$$h_t = \arg\min_{\{F_t^j, \theta_t^j\}} \{E(F_t^j, \theta_t^j)\} \tag{13}$$

As in Eq. (13), the greedy method simultaneously selects the best feature set F_t and learns the parameter θ_t to minimize the overall error.

The comparison of forward selection and greedy training is shown in Fig. 6. In this experiment, 500 eyes data and 2000 non-eyes are used, and the overall accuracy is compared. The greedy method performs much better than the forward selection method at second stage, 95% vs. 83%. With more features added, the benefit of greedy training begins to diminish because AdaBoost algorithm itself can compensate the difference in weight updating of failure samples. This shows that the greedy learning method offers a good performance at the initial stages, which is useful to train the first several layers in a cascade.

5. Experiments

In this section, we evaluate the RNDA features and AdaBoost classifiers on the multi-view face detection as well as on the eye detection.

5.1. Training

In our method, the multi-view faces are roughly divided into five poses: frontal, half left, half right, full left and full right, as shown in Fig. 1. To train a multi-view face detector, face data of multiple poses is collected from various sources, including standard databases, such as FERET

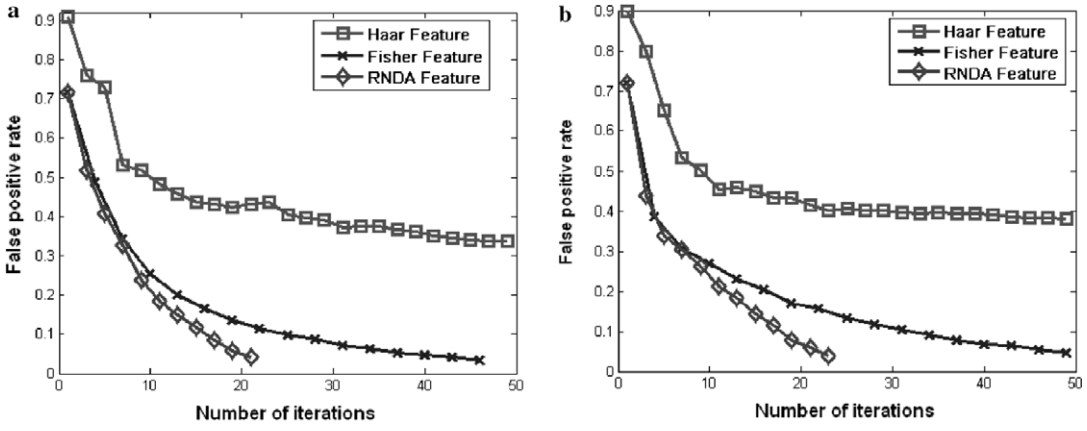

Fig. 5. Comparison of Haar features, Fisher features and RNDA features in AdaBoost. (a) Training results with half profile face data. (b) Training results with full profile face data.

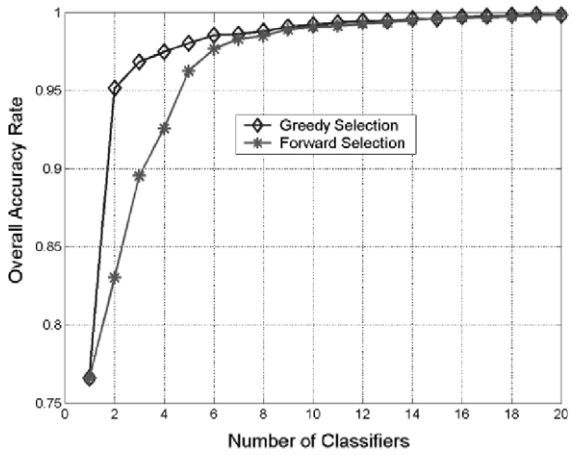

Fig. 6. Greedy training vs. forward selection.

[23], and images collected from the web. All the training data are cropped from images, hand-labeled with pose and slightly perturbed by scaling, shifting and rotation. There are about 10,000 face training samples for each face pose. Some typical face and non-face images used for training are shown in Fig. 7. The non-face images are generated from about 1000 background image. Five hundred pairs of eyes are collected from FERET images [23] and web images to train an eye detector. Some typical eye and non-eye images are shown in Fig. 8.

During training, all collected image patches are normalized to a fixed size, 20 by 20. The intensity of each patch is subtracted by its mean and divided by its standard deviation. The use of integral image allows quickly normalizing the pixel intensity [35]. Then RNDA features are learned from the normalized data. Fig. 9 shows the first several discriminant features for multi-view faces and eyes, where the projection vectors are aligned and equalized for display. All these features describe global characteristics of objects. Some of the characteristics are difficult to be represented with rectangle shapes, such as the pupil, outline of eyes and eyebrows in eyes.

5.2. Frontal and profile face detection

The multi-view face detector has a pyramid structure, which is composed of multiple face detectors under different poses, as shown in Fig. 10. The multi-view face detector contains a frontal face detector and several profile face

Fig. 7. Some face and non-face images used in training. The first two rows show some face images of different poses. The third row shows some non-face images.

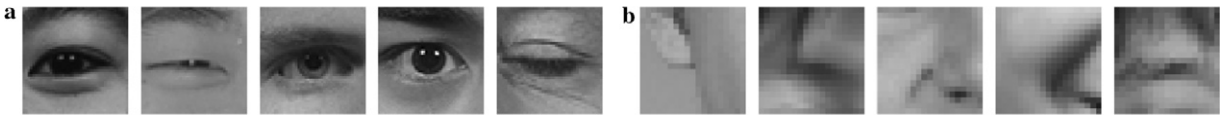

Fig. 8. Some eye and non-eye images used in training. (a) Some eye images. (b) Non-eye training images cropped from face regions.

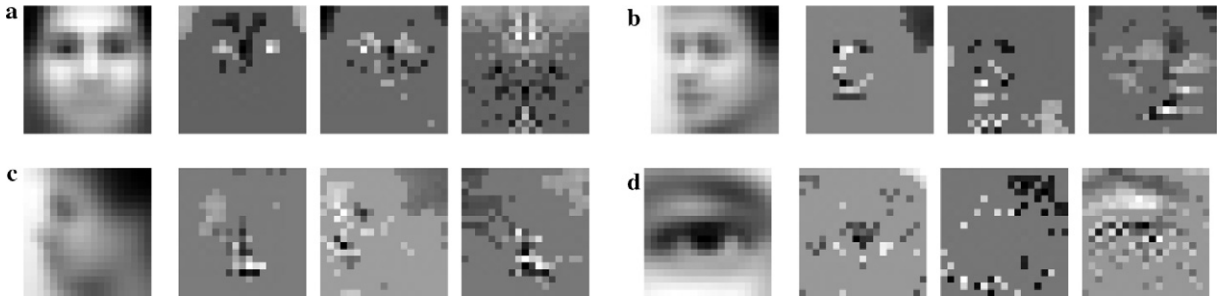

Fig. 9. Good features for multi-view face and eye detection. (a–c) The mean face and first three features for frontal faces, left half and left full profile faces respectively. (d) Mean eye and first three features for eyes.

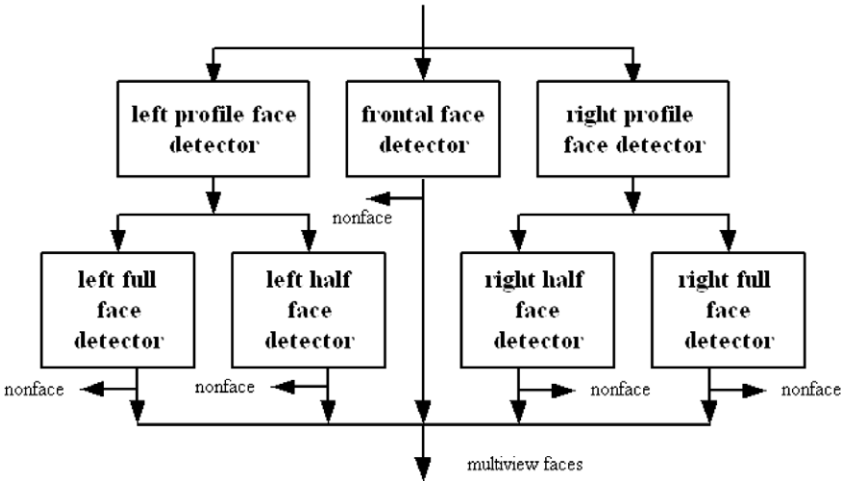

Fig. 10. The pyramid structure of multi-view face detector.

detectors. Due to face symmetry, only left profile face detectors are trained. To find right profile faces, image patches are flipped in detection. There are actually three profile face detectors in the pyramid structure, since the left half and left full profile faces are merged to first train a left profile face detector. To improve the speed in detection, the cascade structure is adopted in each of the face detectors, to quickly discarded the easy-to-classify non-faces [34]. The left profile face detector has 8 layers which can remove about 99.0% of non-face patches in detection. Then, the remaining pathes are sent to the half and full profile face detectors to further remove the non-faces. In detection, image patches at different locations and scales are exhaustively searched since we assume no prior knowledge of face positions in images. Around one real face in an image, each face detector may output overlapped detection results. In our method, we select the detector providing the most detection results, and then average all its overlapped detection results (i.e., the size and positions) to provide a simplified face detection result.

The frontal face detection method is tested on the CMU + MIT test set, which totally has 130 images with 507 frontal faces [25]. The comparison results of Haar features and RNDA features are given in Table 5. The RNDA feature based method can achieve comparable accuracy while with much fewer weak classifiers. Some of the detection results are shown in Fig. 12. With only RNDA features, our frontal face detector runs at 5 frames per second on a P4 2.6G processor. To further improve the speed, Haar features are used to train the first several layers as the frontal end. As a result, the frontal face detector based on such combination also achieve the real time speed (above 15 fps). The related computational issue is discussed at Section 5.4.

Table 5
Comparison results on CMU-MIT set for frontal face detection

Method	Detection rate (%)	# of false	# of weak classifiers
Haar feature	76.1	10	6060 (38 layers)
(Viola and Jones' [34])	91.4	50	
	89.8	65	4297 (32 layers)
RNDA	84.5	13	369
Feature	90.2	64	(nesting structure)

The multi-view face detection method is applied to the CMU profile face test set. The CMU profile face set contains 208 images and 347 profile faces. Compared with frontal face detection result, profile face detection has slightly lower detection rate with a few more false detections. Some detection results are shown in Fig. 12. The ROC curves of Haar features [12] and our RNDA features for profile face detection are shown in Fig. 11. The Haar features implemented by Jones and Viola include both original and extended feature sets. Schneiderman and Kanade's multi-view face detection method achieves better but comparable detection accuracy, with the sacrifice of running speed [30]. Their detector runs at 640 ms per image while our multi-view face detector runs at about 200 ms for each frame.

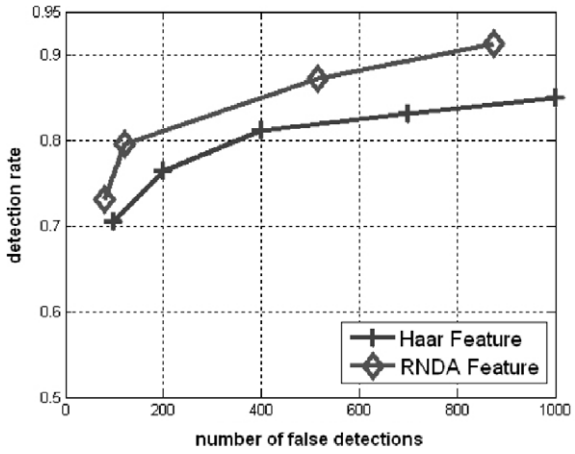

Fig. 11. ROC of profile face detection result on CMU test set. The results of Haar feature are from [12].

5.3. Eye detection

In our method, after a frontal face is detected, eyes are located inside the face region. Some eye detection results are shown in Fig. 13. The eye detection algorithm is validated on about 5600 images in FRGC database V1.0 [22]. The frontal face detection accuracy is around 95.0%. From the detected faces, above 99.0% eyes are successfully detected so that the overall eye detection rate is 94.5%. The absolute pixel errors between the eye center and the ground truth of pupil are given in Table 6 (in pixels). Please note that FRGC images are of high resolution. The average distance between two eyes is about 260 pixels. The comparison of eye localization using Haar features and RNDA

Table 6
Eye localization error on FRGC V1.0

Error	Horizontal		Vertical		Euclidean distance (mean)
	(mean)	(std)	(mean)	(std)	
Pixel	4.9914	4.5808	3.1652	2.6927	6.4016
Normalized (%)	2.04	1.96	1.31	1.35	2.67

Fig. 12. Multi-view face detection results. (a) Frontal face detection results. (b) Multi-view face detection results.

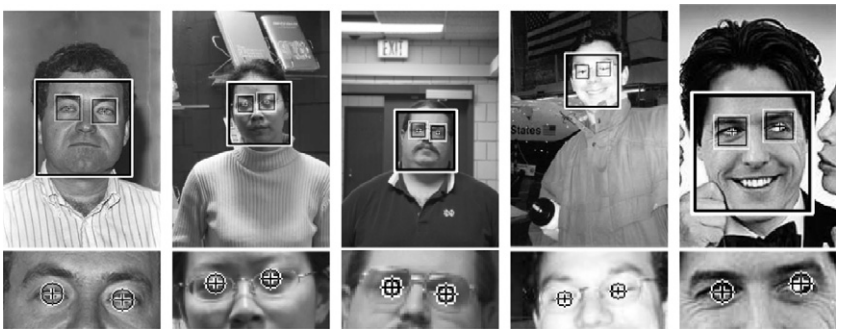

Fig. 13. Eye detection results. First row shows the face and eye detection results. The second row shows enlarged eyes.

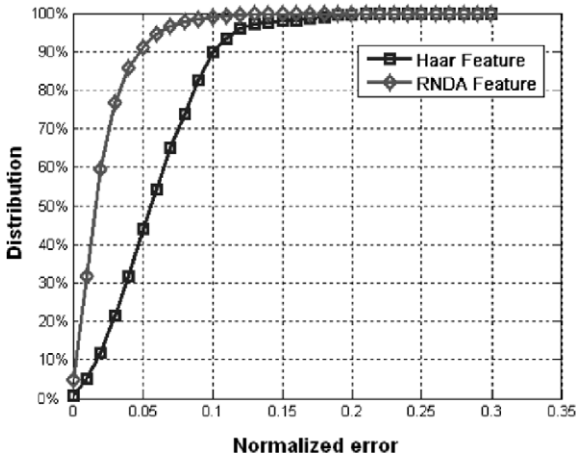

Fig. 14. Comparison of accumulated eye localization errors.

features is shown in Fig. 14. In this figure, the horizontal axis is the normalized localization error, and the vertical axis is the accumulated distribution, which represents the percentage of detected eyes with smaller normalized localization error than the corresponding value at the horizontal axis. From Fig. 14, it is observed that the eye localization based on RNDA features has much smaller localization error than those based on Haar features.

In [36], we further validate the automatic eye localizations by applying them to the face recognition experiments using FRGC V1.0 database sets and PCA and LDA baseline algorithms [22]. Experimental results demonstrate that the face recognition algorithms based on our automatic eye localizations can provide comparable accuracy with those based on manual eye localizations.

5.4. Discussion on computational complexity

Besides accuracy, speed is a very important factor in face detection. Computing discriminant features is more time-consuming than Haar features since it involves dot production. The computational problem is addressed by using Haar features as the front end. The first several cascades are trained with Haar features, and can remove most of non-face patches. The later cascades trained on RNDA features further eliminate the remaining difficult patches. This combination of powerful RNDA features with efficient Haar features leads to improved efficiency for complex object detection (e.g., multi-view faces and eyes) without significantly sacrificing the overall accuracy. As a result, the frontal face detection achieves the real time, and the multi-view face detection runs at 5 frame per second on the 320×240 images on a P4 2.6G processor.

6. Conclusion and future work

This paper presents a novel feature extraction method for multi-view face and eye detection. The RNDA feature extraction method can handle more general class distributions than Fisher discriminant analysis, and can reduce the computational complexity of nonparametric discriminant analysis, so it performs well for the profile face and eye detection. Compared with commonly used Haar features, RNDA features provide better accuracy. Histograms of RNDA features are learned to represent class distributions, then are used to construct probabilistic classifiers. The probabilistic classifiers are further combined with AdaBoost to form a face detector. Experimental results both on the multi-view face detection and on the eye detection demonstrate RNDA feature's success in handling various patterns. Although the discriminant features are more computationally expensive than Haar features, the speed problem can be tackled by using Haar features as frontal end of a cascade. The future work will focus on further improving the speed.

References

[1] M. Bressan, J. Vitria, Nonparametric discriminant analysis and nearest neighbor classification, Pattern Recognition Letters 24 (15) (2003) 2743–2749.
[2] Ian Fasel, Bret Fortenberry, Javier Movellan, A generative framework for real time object detection and classification, Computer Vision and Image Understanding 98 (1) (2005) 182–210.
[3] R.S. Feris, J. Gemmell, K.Toyama, V. Kruger, Hierarchical wavelet networks for facial feature localization, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 118–123.
[4] Yoav Freund, Robert E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: European Conference on Computational Learning Theory, 1995, pp. 23–37.
[5] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, The Annals of Statistics 28 (2) (2000) 337–374.
[6] K. Fukunaga, Nonparametric discriminant analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (1) (1984) 671–678.
[7] Christophe Garcia, Manolis Delakis, Convolutional face finder: A neural architecture for fast and robust face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1408–1423.
[8] B. Heiselet, T. Serre, M. Pontil, T. Poggio, Component-based face detection, CVPR (2001) 657–662.
[9] Rein-Lien Hsu, M. Abdel-Mottaleb, A.K. Jain, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 696–706.
[10] Jeffrey Huang, Harry Wechsler, Eye detection using optimal wavelet packets and radial basis functions (rbfs), International Journal of Pattern Recognition and Artificial Intelligence 13 (7) (1999) 1009–1026.
[11] Qiang JI, Harry Wechsler, Andrew Duchowski, Myron Myron Flickner, Special issue: eye detection and tracking, Computer Vision and Image Understanding (2005) 1–3.
[12] M. Jones, P. Viola, Fast multi-view face detection, Technical Report TR2003-96, MERL, June 2003.
[13] R. Kothari, J.L. Mitchell, Detection of eye locations in unconstrained visual images, ICIP 3 (1996) 519–522.
[14] S.Z. Li, Qingdong Fu, Lie Gu, B. Scholkopf, Yimin Cheng, Hongjiag Zhang, Kernel machine based learning for multi-view face detection and pose estimation, ICCV 2 (2001) 674–679.
[15] S.Z. Li, ZhenQiu Zhang, Floatboost learning and statistical face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1112–1123.

[16] Yongmin Li, Shaogang Gong, H. Liddell, Support vector regression and classification based multi-view face detection and recognition, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 300–305.
[17] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, ICIP 1 (2002) 900–903.
[18] Ce Liu, Hueng-Yeung Shum, Kullback–Leibler boosting, CVPR 1 (2003) 587–594.
[19] Y. Ma., X. Ding, Z. Wang, N. Wang, Robust precise eye location under probabilistic framework, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 339–344.
[20] Baback Moghaddam, Alex Pentland, Probabilistic visual learning for object representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 696–710.
[21] E. Osuna, R. Freund, F. Girosi, Training support vector machines:an application to face detection, CVPR (1997) 130–136.
[22] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, CVPR (2005).
[23] P.J. Phillips, Hyeonjoon Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1090–1104.
[24] D. Roth, M. Yang, N. Ahuja, A snow based face detector, NIPS (2000).
[25] Henry A. Rowley, Shumeet Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.
[26] R.E. Schapire, A brief introduction to boosting, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence, 1999, pp. 246–252.
[27] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.
[28] H. Schneiderman, Feature-centric evaluation for efficient cascaded object detection, CVPR 2 (2004) 29–36.
[29] H. Schneiderman, T. Kanade, A statistical method for 3d object detection applied to faces and cars, CVPR 1 (2000) 746–751.
[30] H. Schneiderman, T. Kanade, Object detection using the statistics of parts, International Journal of Computer Vision 56 (3) (2004) 151–177.
[31] F.Y. Shih, Y. Fu, K. Zhang, Multi-view face identification and pose estimation using b-spline interpolation, Information Sciences 169 (3) (2005) 2005.
[32] Kah-Kay Sung, Tomaso Poggio, Example-based learning for view based human face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 39–51.
[33] R. Verma, C. Schmid, K. Mikolajcayk, Face detection and tracking in a video by propagating detection probabilities, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1216–1228.
[34] Paul Viola, Michael Jones, Robust real-time object detection, International Journal of Computer Vision 57 (2) (2004) 137–154.
[35] Paul, Viola, Michael, Jones, Robust real-time object detection, Interntional workshop on statistical and computational theories of vision, 2001.
[36] Peng Wang, Matthew B. Green, Qiang Ji, James Wayman, Automatic eye detection and its validation, in: IEEE Workshop on Face Recognition Grand Challenge Experiments, 2005.
[37] Peng Wang, Qiang Ji, Multi-view face detection under complex scene based on combined svms, ICPR 4 (2004) 179–182.
[38] Peng Wang, Qiang Ji, Learning discriminant features for multi-view face and eye detection, CVPR 1 (2005) 373–379.
[39] C.A. Waring, X. Liu, Face detection using spectral histograms and svms, IEEE Transactions on Systems, Man and Cybernetics, Part B 35 (3) (2005) 467–476.
[40] Andrew Webb, Statistical pattern recognition, second ed., Wiley, New York, 2002.
[41] Bo Wu, Haizhou Ai, Chang Huang, Shihong Lao, Fast rotation invariant multi-view face detection based on real adaboost, in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 79–84.
[42] Rong Xiao, Ming-Jing Li, Hong-Jiang Zhang, Robust multipose face detection in images, IEEE Transactions on Circuits and Systems for Video Technology 14 (1) (2004).
[43] Kin Choong Yow, R. Cipolla, A probabilistic framework for perceptual grouping of features for human face detection, in: International Conference on Automatic Face and Gesture Recognition, 1996, pp. 16–21.
[44] D. Zhang, S.Z. Li, D. Gatica-Perez, Real-time face detection using boosting in hierarchical feature spaces, ICPR 2 (2004) 411–414.
[45] Z.H. Zhou, X. Geng, Projection functions for eye detection, Pattern Recognition 37 (5) (2004) 1049–1056.