A New Biased Discriminant Analysis Using Composite Vectors for Eye Detection

Chunghoon Kim, *Member, IEEE*, Sang-Il Choi, *Member, IEEE*,
Matthew Turk, *Senior Member, IEEE*, and Chong-Ho Choi, *Member, IEEE*

***Abstract*—We propose a new biased discriminant analysis (BDA) using composite vectors for eye detection. A composite vector consists of several pixels inside a window on an image. The covariance of composite vectors is obtained from their inner product and can be considered as a generalization of the covariance of pixels. The proposed composite BDA (C-BDA) method is a BDA using the covariance of composite vectors. We construct a hybrid cascade detector for eye detection, using Haar-like features in the earlier stages and composite features obtained from C-BDA in the later stages. The proposed detector runs in real time; its execution time is 5.5 ms on a typical PC. The experimental results for the CMU PIE database and our own real-world data set show that the proposed detector provides robust performance to several kinds of variations such as facial pose, illumination, eyeglasses, and partial occlusion. On the whole, the detection rate per pair of eyes is 98.0% for the 3604 face images of the CMU PIE database and 95.1% for the 2331 face images of the real-world data set. In particular, it provides a 99.7% detection rate for the 2120 CMU PIE images without glasses. Face recognition performance is also investigated using the eye coordinates from the proposed detector. The recognition results for the real-world data set show that the proposed detector gives similar performance to the method using manually located eye coordinates, showing that the accuracy of the proposed eye detector is comparable with that of the ground-truth data.**

***Index Terms*—Biased discriminant analysis (BDA), composite feature, composite vector, eye detection, face recognition.**

I. Introduction

RECENTLY, SEVERAL studies have been performed on eye detection as a preprocessing step for face recognition [1]–[8]. After detecting faces in an image, it is necessary to align the faces for recognition. Face alignment is usually performed by warping the image so that the eye positions line up with predefined image coordinates, and the accuracy of the eye coordinates greatly affects the performance of a face recognition system [5], [9], [10]. The location of the eye is commonly measured at the iris or pupil center [11], [12]. According to recent results in the field of face recognition, state-of-the-art methods provide robust performance to several kinds of variations, such as facial expression and illumination, when using manually located eye coordinates [9], [13]. When the eye coordinates were shifted randomly, the face recognition rates degraded rapidly [9], [14], [10]. From these results, we can see that eye detection is very important in face recognition systems. However, as pointed out in [11], eye detection remains a very challenging task due to several variations. The shape and appearance of the eye vary with identity, race, viewing direction, illumination condition, eye motion, etc. Occlusions due to eyeglasses or glare on the glasses can also cause a severe problem [15].

In previous studies, appearance-based methods have been popularly used to discriminate between eyes and noneyes. The appearance-based methods use the photometric appearance as characterized by the pixel intensity distribution or filter responses of the eye and its surroundings [11]. Pentland *et al.* used the eigeneyes, eigennoses, and eigenmouths, based on principal component analysis (PCA), to detect the eyes, nose, and mouth [4]. Huang and Wechsler used wavelet packets for eye representation and radial basis functions for subsequent classification of eyes and noneyes [2]. Ma *et al.* used Haar-like features to detect eyes [3]. Zhu and Ji used a support vector machine to discriminate eyes from noneyes [8]. Wang and Ji constructed a cascade detector for eye detection, using Haar-like features and features obtained from the recursive nonparametric discriminant analysis [6], [7]. Choi and Kim used a cascade detector for eye detection, using features obtained from the modified census transform (MCT) [16]. Song *et al.* also used a cascade detector, employing Haar-like features and rectangle features obtained from a visual-context pattern [17].

In object detection problems such as face and eye detection, a sliding window detection approach [18] is commonly used on an image pyramid because the location and size of the object are unknown. A large number of windows are used for detection, the vast majority of which do not contain target objects. In this case, a cascade detector is an efficient way to detect the object, where a simple classifier with a small number of features is used to reject the majority of detection windows at the first stage [19]–[22]. When constructing a cascade detector, Haar-like features are most frequently used in face and eye detection due to their computational efficiency. Haar-like features use binary rectangles on an image, computing feature values by

Manuscript received April 30, 2011; revised September 2, 2011 and December 28, 2011; accepted January 9, 2012. Date of publication March 6, 2012; date of current version July 13, 2012. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2007-357-D00161).

C. Kim is with the the Qualcomm Research, Seoul 137-920, Korea.

S.-I. Choi is with the Department of Applied Computer Engineering, Dankook University, Gyeonggi-do 448-701, Korea (e-mail: csichoisi@gmail.com).

M. Turk is with the Computer Science Department, University of California, Santa Barbara, CA 93106 USA.

C.-H. Choi is with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-744, Korea.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2186798

1083-4419/$31.00 © 2012 IEEE

summing the pixel values inside the white rectangle(s) and subtracting the pixel values inside the black rectangle(s) [20], [22], [23]. Integral images provide a very efficient method of calculating these features [22]. However, in the later stages of the cascade detector, it is very difficult to reject false positives arising near the eye position because the feature values are so similar. Therefore, more powerful features are essential to remove false positives in the later stages of the cascade detector.

Alternatively, Kim and Choi introduced a new method of extracting composite features for classification problems [9], [24]. In their study, a composite vector consists of a number of primitive variables which correspond to pixels inside a window on an image. The covariance of composite vectors is obtained from their inner product, and a new linear discriminant analysis (LDA) technique is derived by using the covariance of composite vectors. In the derived composite LDA (C-LDA), features are obtained by linear combinations of the composite vectors; these features are called *composite features* because each feature is a vector whose dimension is equal to the dimension of the composite vector. According to their results, composite features by C-LDA showed good performance in classification problems when adjacent primitive variables are strongly correlated as in image data sets or the sonar data set [9], [24]. However, it is inappropriate to apply C-LDA to eye detection directly. C-LDA is an effective method when samples in each class are normally distributed. In eye detection, positive samples for eyes are similar, and they can be assumed to be normally distributed, while negative samples are not. In this case, it is better to use the objective function in biased discriminant analysis (BDA) [25]. BDA was originally proposed for a $(1+x)$-class problem in content-based image retrieval [25]–[27], and recently, several variants of BDA have been proposed to solve the small sample size problem [28], [29] and to improve the retrieval performance by considering the local geometry of positive samples [30]–[32]. BDA tries to find a linear transform that makes the scatter of the positive samples as small as possible while keeping negative samples as far away from the positive samples as possible. BDA does not assume that the negative samples are normally distributed and can be an appropriate method in detection problems as well as in image retrieval.

In this paper, we propose a new BDA approach using composite vectors. We call this approach composite BDA (C-BDA). C-BDA is derived from BDA by using the covariance of composite vectors instead of the covariance of pixels. To detect the eye coordinates in a face image, we construct a hybrid cascade detector. At the earlier stages in the hybrid cascade detector, Haar-like features are used to remove the majority of noneyes. At the later stages, composite features obtained from C-BDA are used to discriminate between eyes and noneyes, which are difficult to discriminate by Haar-like features. The experimental results for the CMU PIE database [33] and the real-world data set show that the proposed detector provides robust performance to several kinds of variations such as facial pose, illumination, glasses, and partial occlusion. On the whole, the detection rate per pair of eyes is 98.0% for the 3604 face images of the CMU PIE database and 95.1% for the 2331 face images of the real-world data set. Face recognition performance is also investigated on the real-world data set using the eye coordinates from the proposed detector. The recognition results for the real-world data set show that the proposed detector gives better performance than other methods and gives similar performance to the method using manually located eye coordinates. This implies that the accuracy of the proposed eye detector is comparable with that of the ground-truth data.

In the following section, we explain how patterns are represented using composite vectors and derive C-BDA. Experimental results are described in Section III, and conclusions follow in Section IV. A portion of this paper was presented in [15], and the differences between this paper and [15] are as follows: 1) More details of the proposed C-BDA method are included, and the posterior probability of the C-BDA feature space is investigated in Section II; 2) a glare reduction technique in order to minimize the influence of glare on glasses is proposed, and more explanations on the proposed cascade detector are added in Section III-A; and 3) experimental results on eye detection and face recognition using the real-world data set are added in Section III-C.

II. BDA Using Composite Vectors

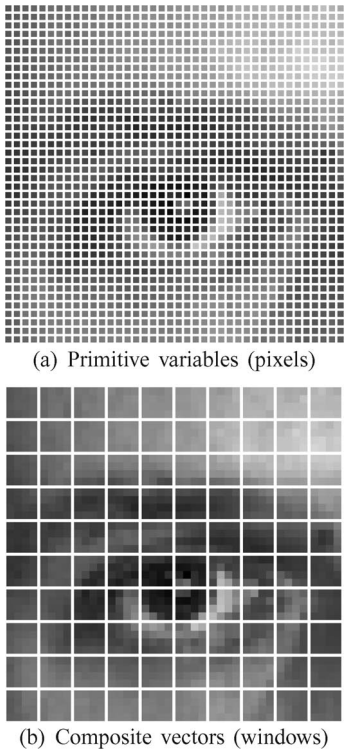
In this section, we first define composite vectors and their covariance in an image. Then, we derive C-BDA which is a BDA using the covariance of composite vectors. Unlike [24], we differentiate the composite feature from the composite vector in order to avoid confusion in terminology.

A. Composite Vectors and Their Covariance

In general, a pattern is represented by a set of variables, which are called primitive variables [24]. In appearance-based models, the intensity of each pixel in an image is used as a primitive variable. Conventional methods such as PCA and LDA use the covariance of pixels, i.e., the covariance of primitive variables. Fig. 1(a) shows an example of the primitive variables. The size of the image is 40 $\times$ 40 (pixels), and thus, there are 1600 primitive variables. As can be seen in this figure, neighboring pixels are highly correlated. If we use the covariance of these pixels, there is a huge number of combinations $({}_{1600}C_2 = 1\,279\,200)$. Since there are high correlations between neighboring pixels, it is redundant to use all of these 1 279 200 combinations.

Let us consider a composite vector composed of a number of primitive variables and the covariance of composite vectors instead of the covariance of primitive variables. Table I shows a summary of variables, vectors, and matrices used in Section II. Let $A \in \mathbb{R}^{a_r \times a_c}$ denote an image, where a_r and a_c are the height and width of the image. Let $\mathbf{H}$ denote a set of windows $\{H_1, H_2, \ldots, H_n\}$ in the image. Each window $H_i \in \mathbb{R}^{h_r \times h_c}$ has $l(= h_r \times h_c)$ pixels, where h_r and h_c are the height and width of the window. If there is no overlap of windows, the number of windows n is p/l where p is the total number of pixels in A. Obviously, more windows can be obtained if neighboring windows overlap each other. Then, the composite vector $\mathbf{x}_i$ is obtained from H_i as

$$\mathbf{x}_i = \mathcal{O}_{\mathcal{L}}(H_i) \tag{1}$$



(a) Primitive variables (pixels)

(b) Composite vectors (windows)

Fig. 1. Primitive variables versus composite vectors: Size of each image is 40×40 (pixels), and the window sizes of (a) and (b) are 1×1 and 4×4, respectively. (a) Primitive variables (pixels). (b) Composite vectors (windows).

TABLE I
SUMMARY OF VARIABLES, VECTORS, AND MATRICES USED IN SECTION II

symbol	dimension	definition
A	$\mathbb{R}^{a_r \times a_c}$	image having $a_r \times a_c$ pixels
H_i	$\mathbb{R}^{h_r \times h_c}$	window defined on A, having $h_r \times h_c$ pixels ($i = 1, \ldots, n$)
$\mathbf{x}_i$	$\mathbb{R}^l$	composite vector obtained from H_i ($i = 1, \ldots, n$)
l	1	size of the composite vector ($l = h_r \times h_c$)
n	1	number of the composite vectors in A
X	$\mathbb{R}^{n \times l}$	set of composite vectors in A
$\boldsymbol{\chi}_j$	$\mathbb{R}^n$	column vector of X, subsampled version of A ($j = 1, \ldots, l$)

where $\mathcal{O}_{\mathcal{L}}(\cdot)$ is the lexicographic ordering operator that transforms a matrix into a vector by ordering the rows of the matrix one after the other [34]. Therefore, $\mathbf{x}_i$ becomes an l-dimensional vector. Fig. 1(b) shows an example of the composite vectors. In this image, 4×4 windows are used for making composite vectors, and there are 100 composite vectors in total.

The covariance of composite vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as [24]

$$c_{ij} = E\left[(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T (\mathbf{x}_j - \bar{\mathbf{x}}_j)\right], \qquad i, j = 1, 2, \ldots, n \quad (2)$$

where $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ are the mean vectors of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Note that c_{ij} corresponds to the total sum of covariances between the corresponding pixels in H_i and H_j. It contains information on statistical dependence among multiple pixels. When l is 1, $\mathbf{x}_i$ becomes a scalar, and c_{ij} in (2) is the same as the covariance of primitive variables. Therefore, the covariance of composite vectors can be considered as a generalization of the covariance of primitive variables.

Let C denote a covariance matrix based on the composite vectors. For a sample set of N images, the covariance matrix C is computed as

$$C = \frac{1}{N} \sum_{k=1}^{N} (X(k) - M)(X(k) - M)^T \quad (3)$$

where $X(k) = [\mathbf{x}_1(k) \ldots \mathbf{x}_n(k)]^T$ for the kth sample and $M = [\bar{\mathbf{x}}_1 \ldots \bar{\mathbf{x}}_n]^T$. Note that $X(k) \in \mathbb{R}^{n \times l}$ and $C \in \mathbb{R}^{n \times n}$.

B. Analysis of the Covariance Matrix Based on Composite Vectors

We now investigate the covariance matrix C in (3), based on composite vectors, comparing it with the covariance matrix based on primitive variables. Let $\boldsymbol{\chi}_j(k), \mathbf{m}_j \in \mathbb{R}^n$ denote the column vectors of $X(k)$ and M, respectively. Then, $X(k) = [\boldsymbol{\chi}_1(k) \ldots \boldsymbol{\chi}_l(k)]$, and $M = [\mathbf{m}_1 \ldots \mathbf{m}_l]$. We rewrite (3) as

$$C = \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{l} \left(\boldsymbol{\chi}_j(k) - \mathbf{m}_j\right)\left(\boldsymbol{\chi}_j(k) - \mathbf{m}_j\right)^T. \quad (4)$$

The number of outer products of vectors in (4) is Nl, which is l times larger than that in the covariance matrix based on primitive variables. This is because $X(k)$ has l column vectors of $\boldsymbol{\chi}_j(k)$. Fig. 2 shows l images of $\boldsymbol{\chi}_j(k)$ for some k. Since $\boldsymbol{\chi}_j(k)$ takes the jth element from each of the n windows, it is a subsampled version of the image $A(k)$. In this case, l and n are 16 and 361 (19×19), respectively, because the 4×4 windows are used for making the composite vectors in the 40×40 image and they overlap either horizontally or vertically by 50%. For the purpose of visualization, $\boldsymbol{\chi}_j(k)$'s are represented as images. As can be seen in Fig. 2, there is a small variation in eye positions of $\boldsymbol{\chi}_j(k)$ images. All these 16 images are used to compute C in (4) as if they were individual samples. If l becomes larger, more images with a larger variation are used for computing C.

From this perspective, the proposed approach is similar to the traditional approach using l times more images of smaller size. The difference is $\mathbf{m}_j$ in (4), which is dependent on j unlike in the covariance matrix based on primitive variables. Refer to [35] for further details on the difference between the composite vector-based and primitive variable-based covariance matrices.

Let us further investigate C in (4). We rewrite (4) as

$$C = \sum_{j=1}^{l} S_j \quad (5)$$

where $S_j = (1/N) \sum_{k=1}^{N} (\boldsymbol{\chi}_j(k) - \mathbf{m}_j)(\boldsymbol{\chi}_j(k) - \mathbf{m}_j)^T$. Here, S_j is the primitive covariance matrix which corresponds to the covariance matrix based on primitive variables. Since

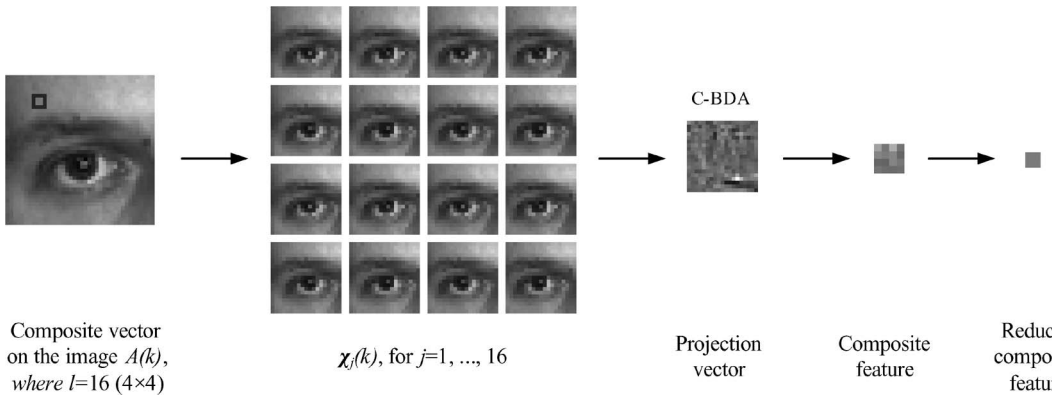


Fig. 2. Schematic diagram of C-BDA: In the leftmost image, the size of image $A(k)$ is 40 × 40, and a 4 × 4 window is overlaid on the image. Since the 4 × 4 windows are used for making the composite vectors and they overlap either horizontally or vertically by 50%, the size of the composite vector l is 16, and the number of composite vectors n is 361 (19 × 19). In the second column, $\boldsymbol{\chi}_j(k) \in \mathbb{R}^{361}$ $(j = 1, \ldots, 16)$ is represented as a 19 × 19 image for visualization. The projection vector is obtained by C-BDA and is represented as a 19 × 19 image in the figure. The composite feature in the figure is obtained by projecting $\boldsymbol{\chi}_j(k)$'s onto the projection vector. Thus, the composite feature is a 16-D vector as the composite vector and is represented as a 4 × 4 image in the figure. Finally, the composite feature is further reduced by applying a downscaling operator. In this case, the downscaling factor r is set to 16, so the 4 × 4 elements of the composite feature are represented by their average value.

C can be represented as the sum of l S_j's, it is a composite of primitive covariance matrices. This is due to the definition of the covariance of composite vectors in (2), where the covariance c_{ij} is defined as the sum of l covariances between the corresponding pixels in $\mathbf{x}_i$ and $\mathbf{x}_j$.

C. BDA Using the Covariance of Composite Vectors (C-BDA)

The composite vectors and their covariance were defined in Section II-A. In the following, a new discriminant analysis approach using the covariance of composite vectors is derived for eye detection. In eye detection, positive samples for eyes are similar, and they can be assumed to be normally distributed, while negative samples are not. In this case, it is better to use the objective function in BDA [25]. BDA tries to find a linear transform that makes the scatter of the positive samples as small as possible while keeping negative samples as far away from the positive samples as possible. BDA does not assume that the negative samples are normally distributed and can be an appropriate method in detection problems.

Let us now derive C-BDA, which is a BDA using the covariance of composite vectors. In eye detection, $X(k) = [\mathbf{x}_1(k) \ldots \mathbf{x}_n(k)]^T$ is either a positive or negative sample. Let $X_P(k) \in \mathbb{R}^{n \times l}$ and $X_N(k) \in \mathbb{R}^{n \times l}$ denote the sets of composite vectors of the kth positive sample and the kth negative sample, respectively. In C-BDA, the set of projection vectors W_B is obtained by

$$W_B = \arg\max_W \frac{|W^T C_N W|}{|W^T C_P W|} \tag{6}$$

where $W_B = [\mathbf{w}_1 \ldots \mathbf{w}_m] \in \mathbb{R}^{n \times m}$, and the covariance matrices $C_P \in \mathbb{R}^{n \times n}$ and $C_N \in \mathbb{R}^{n \times n}$ are defined as

$$C_P = \frac{1}{N_P} \sum_{k=1}^{N_P} (X_P(k) - M_P)(X_P(k) - M_P)^T, \tag{7}$$

$$C_N = \frac{1}{N_N} \sum_{k=1}^{N_N} (X_N(k) - M_P)(X_N(k) - M_P)^T. \tag{8}$$

Here, $M_P = (1/N_P) \sum_{k=1}^{N_P} X_P(k)$ is the mean of the positive samples, and N_P and N_N are the number of positive and negative samples, respectively. C_P and C_N correspond to the covariance matrices of positive and negative samples, respectively. The optimization problem of (6) can be computed in two steps as in C-LDA [24]. First, C_P is transformed to an identity matrix by $\Psi\Theta^{-(1/2)}$, where Ψ and Θ are the eigenvector and diagonal eigenvalue matrices of C_P, respectively. Let C'_P and C'_N denote the covariance matrices of positive and negative samples after whitening, respectively. Then, $C'_P = I$, and $C'_N = (\Psi\Theta^{-(1/2)})^T C_N (\Psi\Theta^{-(1/2)})$. Second, C'_N is diagonalized by Φ, which is the eigenvector matrix of C'_N. Therefore, W_B is expressed with m column vectors of $\Psi\Theta^{-(1/2)}\Phi$, corresponding to the m largest eigenvalues of C'_N. Note that $C''_P = I$ and $C''_N = (\Psi\Theta^{-(1/2)}\Phi)^T C_N (\Psi\Theta^{-(1/2)}\Phi)$, where C''_P and C''_N denote the covariance matrices of positive and negative samples after diagonalization of C'_N, respectively. In summary, C-BDA finds a linear transform by which the covariance matrix of positive samples becomes an identity matrix and negative samples are as far away from the mean of the positive samples as possible.

Once W_B is determined, the set of features $Y(k)$ is obtained from $X(k)$ as

$$Y(k) = W_B^T X(k), \qquad k = 1, 2, \ldots, N \tag{9}$$

where $Y(k) \in \mathbb{R}^{m \times l}$ has m features $[\mathbf{y}_1(k) \ldots \mathbf{y}_m(k)]^T$. Here, the feature $\mathbf{y}_i(k) \in \mathbb{R}^l$ is called a composite feature because it is a vector obtained from the linear combination of composite vectors in $X(k)$ from $\mathbf{x}_1(k)$ to $\mathbf{x}_n(k)$. Since C-BDA with $l = 1$ is the same as BDA, C-BDA can be considered as a generalization of BDA.

The first projection vector $\mathbf{w}_1$ is represented in Fig. 2. In this case, n is 361, and $\boldsymbol{\chi}_j(k)$'s are 361-D vectors. Therefore, $\mathbf{w}_1$ is a 361-D vector and is represented as a 19 × 19 image in the figure. The composite feature in the figure is obtained by projecting $\boldsymbol{\chi}_j(k)$'s onto the projection vector. Thus, the composite feature is a 16-D vector as the composite vector

and is represented as a 4×4 image in the figure. Since the differences between adjacent $\boldsymbol{\chi}_j(k)$ images are very small, the correlations between adjacent elements of the composite feature are very strong. This coincides with the analysis in [35]. Therefore, the dimension of the composite feature can be reduced significantly. Let l_r denote (l/r). Then, $Y(k)$ becomes

$$Z(k) = [\mathbf{z}_1(k) \dots \mathbf{z}_m(k)]^T \tag{10}$$

where $\mathbf{z}_i(k) = \mathcal{O}_{\mathcal{L}}[\mathcal{O}_{\mathcal{D}(\mathrm{r})}\{\mathcal{O}_{\mathcal{L}}^{-1}(\mathbf{y}_i(k))\}] \in \mathbb{R}^{l_r}$, $(i = 1, \dots, m)$. Here, $\mathcal{O}_{\mathcal{D}(\mathrm{r})}$ is a downscaling operator with factor r, where r elements are represented by their average value. As a result of $\mathcal{O}_{\mathcal{D}(\mathrm{r})}(\cdot)$, the number of elements in $\mathcal{O}_{\mathcal{L}}^{-1}(\mathbf{y}_i(k))$ is reduced from l to l_r. In Fig. 2, the downscaling factor r is set to 16, so the 4×4 elements of each composite feature are represented by their average value. Note that $Z(k)$ corresponds to the set of reduced composite features of the kth image.

If r is appropriately chosen, the reduced composite feature can be obtained directly without projecting all the $\boldsymbol{\chi}_j(k)$'s onto the projection vector. For example, the reduced composite feature in Fig. 2 is obtained by projecting only the mean of the 16 $\boldsymbol{\chi}_j(k)$ images onto the projection vector. Therefore, the computation time to obtain composite features can be reduced significantly.

D. Posterior Probability in the Feature Space of C-BDA

In the previous section, we derived C-BDA using the covariance of composite vectors for eye detection. For a given sample $X(k) = [\mathbf{x}_1(k) \dots \mathbf{x}_n(k)]^T \in \mathbb{R}^{n \times l}$, we obtained the set of reduced composite features $Z(k) = [\mathbf{z}_1(k) \dots \mathbf{z}_m(k)]^T \in \mathbb{R}^{m \times l_r}$ by C-BDA. Let us investigate the posterior probability of $Z(k)$, which is a good indicator to decide whether the given sample $X(k)$ belongs to the positive class or not. Let c_P and c_N denote the positive class and negative class, respectively. Let us focus on $p(c_P|Z(k))$ because $p(c_N|Z(k))$ can be easily computed from $p(c_P|Z(k))$ by using $p(c_N|Z(k)) = 1 - p(c_P|Z(k))$. As mentioned in Section II-C, the covariance matrix of positive samples C_P'' becomes an identity matrix in the C-BDA feature space. Considering that positive samples for eyes are similar, we can assume that the positive samples are normally distributed in the C-BDA feature space.

We first derive the posterior probability of $Z(k)$ when $l_r = 1$. In this case, the conditional probability $p(Z(k)|c_P)$ can be obtained as

$$\begin{aligned} p\left(Z(k)|c_P\right) &= \frac{1}{(2\pi)^{m/2}\left|C_P''\right|^{1/2}} \\ &\quad \times exp\left(-\frac{1}{2}\left(Z(k)-M_P''\right)^T C_P''\left(Z(k)-M_P''\right)\right) \\ &= \frac{1}{(2\pi)^{m/2}} exp\left(-\frac{1}{2} d_E^2\left(Z(k), M_P''\right)\right) \end{aligned} \tag{11}$$

where M_P'' is the mean of positive samples $Z(k)$ and $d_E(Z(k), M_P'')$ is the Euclidean distance between $Z(k)$ and M_P''. In (11), we can see that $p(Z(k)|c_P)$ is inversely proportional to the Euclidean distance between $Z(k)$ and M_P''. Since $p(c_P|Z(k)) = p(c_P)p(Z(k)|c_P)/p(Z(k))$ and $p(c_P)$ and $p(Z(k))$ can be considered as constants for each sample $Z(k)$, the posterior probability $p(c_P|Z(k))$ can be represented as

$$p\left(c_P|Z(k)\right) \propto exp\left(-\frac{1}{2} d_E^2\left(Z(k), M_P''\right)\right). \tag{12}$$

Now let us derive the posterior probability of $Z(k)$ when $l_r > 1$. In this case, each composite feature $\mathbf{z}_i(k)$ $(i = 1, \dots, m)$ of $Z(k)$ is l_r-dimensional vector. Then, the Euclidean distance between $Z(j)$ and $Z(k)$ can be defined as [9]

$$d_E\left(Z(j), Z(k)\right) = \left\{\sum_{i=1}^{m} \left(\left\|\mathbf{z}_i(j) - \mathbf{z}_i(k)\right\|_2\right)^2\right\}^{\frac{1}{2}} \tag{13}$$

where $\|\cdot\|_2$ is the 2-norm. From this definition, the posterior probability $p(c_P|Z(k))$ can be computed in the C-BDA feature space and is also inversely proportional to $d_E(Z(k), M_P'')$ as in (12). Consequently, when T_H is a predefined threshold, the given sample $X(k)$ is classified as an eye if $d_E(Z(k), M_P'') < T_H$.

III. Experimental Results

This section describes our proposed detector using composite features and experimental results for eye detection and face recognition.

A. Training of the Hybrid Cascade Detector

For training the detector, 1800 images were collected from the CMU PIE database [33], the BioID database [36], and our own collected database of face images, which includes several kinds of variations such as pose, illumination, glasses, partial occlusion, closed eyes, and low resolution. From these images, we first located a face in an image by using a face detector based on the MCT [37]. Since faces are generally symmetric, only a right eye detector is trained as in [10] and [17], and a left eye is found on a vertically inverted image by using the right eye detector. The positive samples for training are obtained from right eyes, and the negative samples are randomly chosen from the other regions of face images. Some positive and negative samples are shown in Fig. 3(a) and (b), respectively. Each positive sample is cropped in proportion to the interocular distance (distance between the two eyes), where the eye coordinates are measured at the center of the iris. The cropping window is a square whose side length is 0.76 times the interocular distance and is rescaled to a size of 18×18 (pixels).

In [15], the proposed detector showed a 99.4% detection rate for the 2120 images without glasses while it showed 93.5% for the 1484 images with glasses. As pointed out in [15], incorrect detections of eyes are mainly caused by glare on glasses. Therefore, glare reduction is essential to robust eye detection. First, we investigated pixel values with glare in face images. Fig. 4 shows two probability distributions, where one is obtained from pixel values with glare and the other is from pixel values

Fig. 3. Eye and noneye samples used for training. (a) Positive samples. (b) Negative samples at the first stage. (c) Negative samples at the fifth stage.

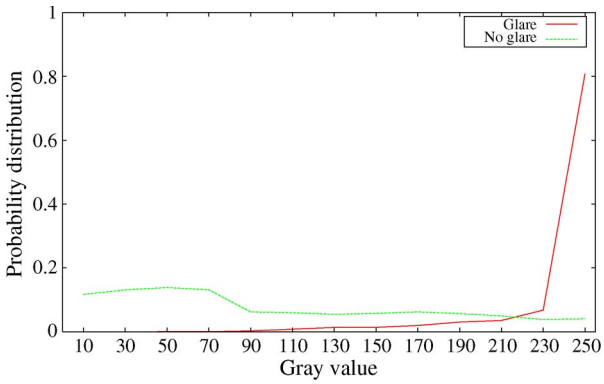


Fig. 4. Two probability distributions of the pixel gray values with and without glare.

without glare. In the figure, the horizontal axis denotes the pixel gray values, and the vertical axis denotes the probability for each gray value. As can be seen in the figure, pixels without glare are widely distributed over the range between 0 and 255, whereas pixels with glare are concentrated near the maximum value of 255. In order to alleviate the influence of glare, we cap the image pixel values at T_G, i.e., pixel values larger than T_G are changed into T_G. In the following experiments, T_G is set to 210 for glare reduction.

After applying the glare reduction technique, intensity normalization is also applied to each detection window to minimize the effect of different light conditions. For intensity normalization, pixels in each detection window are normalized to have zero mean and unit variance [22]. The mean and variance of each detection window in an image can be easily computed using two integral images obtained from the original image and its squared image, as in [20].

When detecting the eye coordinates in a face image, a sliding window detection approach is used on an image pyramid [18]. In the image pyramid, four levels of images, 80 $\times$ 49 (width $\times$ height), 67 $\times$ 41, 56 $\times$ 34, and 47 $\times$ 28, are sequentially obtained from the face image. A detection window of 18 $\times$ 18 (pixels) is then scanned across each image in the image pyramid. A large number of detection windows are used for detection, the vast majority of which do not contain eyes. In this case, a cascade detector is an efficient way to detect the eyes [20], [22]. The structure of the cascade detector is a degenerate decision tree [38]–[40]. Fig. 5 shows the schematic diagram of the cascade detector. The classifier in each stage aims to reject a certain fraction of false positives while correctly detecting almost all true positives. In our case, each stage is trained to reject more than 50% of the false positives while correctly detecting more than 99% of the true positives. At the first stage, a simple classifier with a small number of features is used to reject the majority of detection windows. Those windows which are not rejected by the first classifier are processed by a sequence of classifiers. If any classifier rejects a detection window, no further processing is performed for that window. This is a very efficient way to remove false positives in terms of speed.

When training a cascade detector, the boosting technique is typically used [20], [22]. From the second stage of the cascade detector, the negative samples are obtained from the false positives of the previous stage. That means the classifier of the current stage focuses more on the detection windows that are misclassified at the previous stage. The set of positive samples are also a little bit different from the set of positive samples in the previous stage. As the positive and negative samples vary with the stages, features obtained in each stage also vary, and each stage has its own discriminative power. This provides a very efficient way to remove false positives while correctly detecting true positives, in terms of accuracy.

Haar-like features and composite features can be used for classifiers to discriminate between eyes and noneyes. Haar-like features are very efficient in computation [22], while composite features carry powerful discriminative information. The advantages of these two features can be combined by using a hybrid cascade detector. At the earlier stages in the detector, Haar-like features are used to remove a majority of the noneyes. At the later stages, composite features obtained by C-BDA are used to discriminate between eyes and noneyes in regions which are difficult for Haar-like features to discriminate.

Fig. 6(a) shows the receiver operating characteristic (ROC) curves on the validation set of the first stage. At the first stage of the hybrid cascade detector, 1200 images randomly chosen from the 1800 images are used for training, and the remaining 600 images are used for validation. In the training set, there are 1200 positive samples obtained from right eyes and 2400 negative samples obtained from the other regions of images. In the validation set, there are 600 positive and 6000 negative samples; more negative samples are used to make the validation phase similar to a real detection situation. In the figure, the false positive rate on the horizontal axis corresponds to the number of noneyes classified as eyes among 6000 noneye samples, and the correct detection rate corresponds to the number of eyes classified as eyes among 600 eye samples. In C-BDA, a sample is classified as an eye if the distance from the mean of positive samples is smaller than a predefined threshold. As the threshold increases, the detection rate and false positive rate increase. In this experiment, the results are obtained by using six features of each method. As can be seen in Fig. 6(a), when the false positive rate is 40%, the detection rates of Haar-like, BDA, and C-BDA features are 99.3%, 99.3%, and 99.8%, respectively.

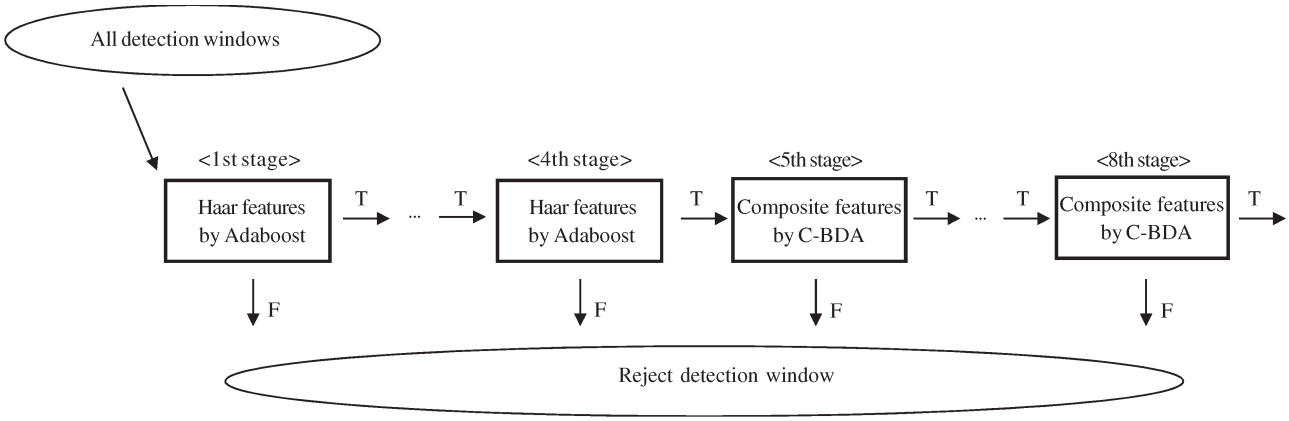


Fig. 5. Schematic diagram of the hybrid cascade detector.

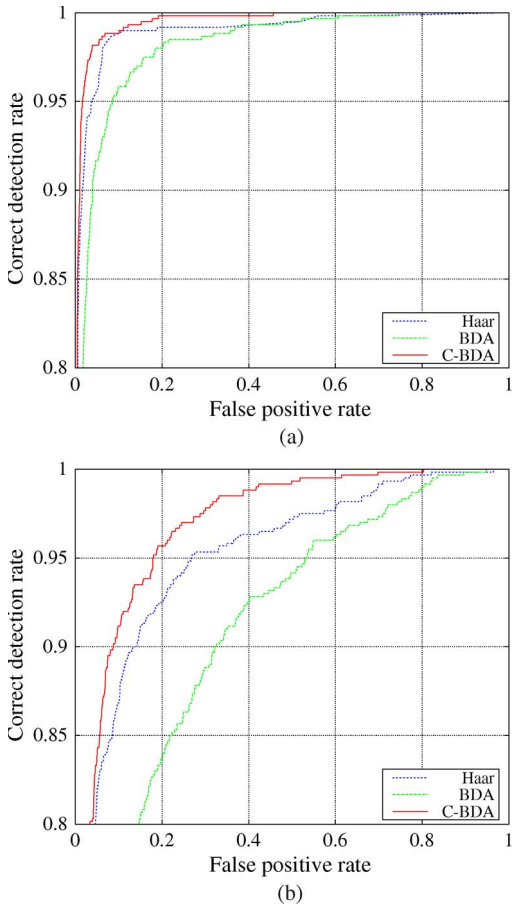


Fig. 6. ROC curves comparing C-BDA features with Haar-like features at the first and fifth stages of the cascade detector. (a) First stage. (b) Fifth stage.

Since Haar-like features require less computation than the other features, it is efficient to use Haar-like features at the first stage. In this way, the first four stages of the cascade detector are constructed using 6, 20, 46, and 94 Haar-like features, respectively, and the next four stages are constructed using C-BDA features, as shown in Fig. 5.

Fig. 7 shows the detection results of the cascade detector on a training image. The red and yellow points on each image are the pixels classified as eyes at the current stage of the cascade detector, where the red and yellow correspond to true positives and false positives, respectively. Fig. 7(a) shows the candidate positions for right and left eyes. These positions are obtained by analyzing the face detection result for the training images. Fig. 7(b)–(e) show the detection results from the first stage to the fourth stage. For 1800 images in the training and validation sets, 79.6% of detection windows are removed at the first stage of the cascade detector. After the first stage, the number of remaining windows on average is 384.1, where 131.4 and 252.7 windows contain eyes and noneyes, respectively.

Although the Haar-like features are easy to compute, they have limited discriminative power [3], [7]. Particularly at the later stages in the cascade detector, the Haar-like features cannot remove false positives efficiently while correctly detecting eyes. Some negative samples at the fifth stage are shown in Fig. 3(c), where they were obtained from false positives of the previous stage. As can be seen in the figure, most of negative samples are obtained near the eye position of face images. Fig. 6(b) shows the ROC curves on the validation set of the fifth stage. In C-BDA, the 3×3 $(h_r = 3, h_c = 3)$ windows are used for making the composite vectors in an 18×18 image, and the downscaling factor r is set to 9 (3×3), i.e., nine elements of each composite feature are represented by their average value. When the false positive rate is 40%, the C-BDA features provide a detection rate of 98.8%, which is 2.5% and 6.3% higher than those of Haar-like and BDA features, respectively. From this result, we can see that the C-BDA features are more effective in later stages than the Haar-like and BDA features. In this way, the later four stages of the cascade detector are constructed using C-BDA features, as shown in Fig. 5. Fig. 7(f)–(i) show the detection results from the fifth stage to the eighth stage. At the eighth stage, the number of remaining windows on average is 55.7, where 52.7 and 3.0 windows contain eyes and noneyes, respectively.

After the eighth stage of the cascade detector, multiple detections usually occur around each eye. In this case, overlapping detection windows are combined into one [22]. If more than one detection is made by this combining procedure, the most probable one is selected as the final result in the right and the left part of a face. Fig. 7(j) shows the final detection result of the proposed cascade detector.

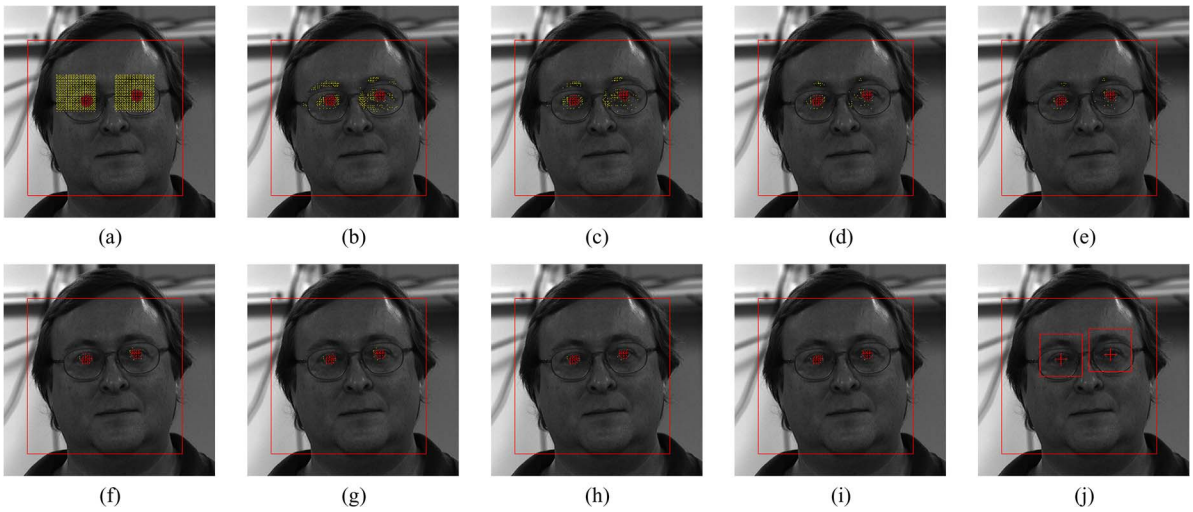


Fig. 7. Eye detection results of the cascade detector on a training image: Red and yellow points on each image correspond to true positives and false positives, respectively. (a) Candidates for eyes. (b) Stage 1. (c) Stage 2. (d) Stage 3. (e) Stage 4. (f) Stage 5. (g) Stage 6. (h) Stage 7. (i) Stage 8. (j) Final result.

B. Test Results for the CMU PIE Database

The CMU PIE database [33] was used to evaluate the robustness of the proposed detector to glasses, pose, illumination, and closed eyes. In this experiment, we tested 3604 face images of 68 people, none of which were used for training. Among 68 people, 28 are wearing glasses, and 40 are not. There are eight images per person with four different poses, named "S(B)_05," "S(B)_07," "S(B)_09," and "S(B)_29," where eyes are open and closed in the "S" and "B" images, respectively. The images named "05" and "29" are right and left 3/4-profile images, respectively, and there are up and down variations in the images named "07" and "09", respectively. There are two sets of 24 images per person under different illumination conditions, named from "27_00" to "27_23" and "05_00" to "05_23," where "27" and "05" mean a frontal and a 3/4-profile image, respectively. Excluding three images of "27_02," "27_08," and "27_16" used for training, there are 21 frontal and 24 3/4-profile images of each person under different illumination conditions.

In order to differentiate between true and false detections, we define the normalized error as follows. Let d_{lr} denote the interocular distance in pixels. Let e_l and e_r denote the Euclidean distances between manually and automatically located coordinates of the left and the right eye, respectively. Then, the normalized error e_n is computed as

$$e_n = \frac{\max(e_l, e_r)}{d_{lr}}. \tag{14}$$

We consider a detection result as correct if $e_n \leq k_e$, where k_e is the threshold of the normalized error.

Table II shows the detection results with respect to several variations, where k_e is set to 0.125 as in [5]. At the first column, the pose variation means an out-of-plane rotation such as left, right, up, and down, and "open" and "closed" mean open and closed eyes, respectively. As can be seen in the table, the proposed detector gives robust performance to several kinds of variations such as facial pose, illumination, and closed eyes. It provides a 99.7% detection rate for the images without glasses.

TABLE II
EYE DETECTION RESULTS FOR THE CMU PIE DATABASE

variation	w/o glasses	with glasses	average
pose (open)	100%	98.2%	99.3%
pose (closed)	100%	89.3%	95.6%
illum. (frontal)	100%	96.8%	98.7%
illum. (3/4 profile)	99.4%	95.1%	97.6%
total	99.7%	95.6%	98.0%

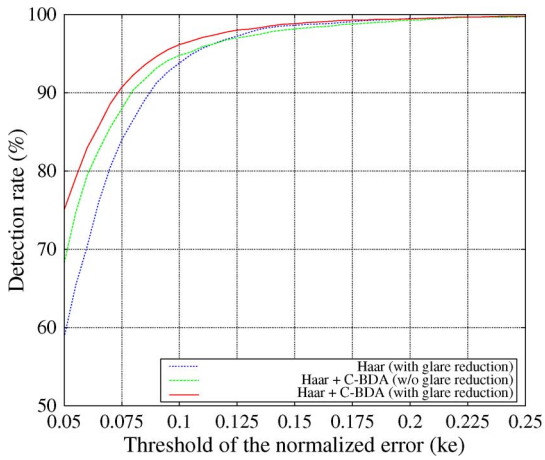


Fig. 8. Detection rates with respect to the threshold of the normalized error for the CMU PIE database.

When k_e is 0.125, 0.15, and 0.20, the total detection rates are 98.0%, 98.8%, and 99.4%, respectively.

Fig. 8 shows the eye detection results with respect to the threshold of the normalized error for the CMU PIE database. As shown in the figure, the proposed detector using both Haar-like and C-BDA features gives better performance than the detector using only Haar-like features. In particular, if the thresholds are 0.05 and 0.1, the proposed detector gives detection rates of 75.0% and 96.1%, which are 16.3% and 2.4% higher than those of the detector using Haar-like features, respectively. This means that C-BDA features enable more accurate eye detection.

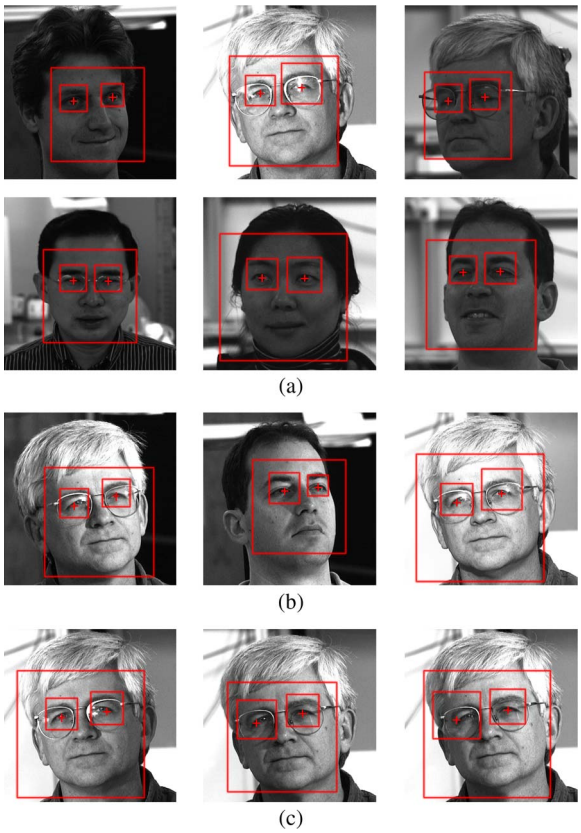


Fig. 9. Examples of the correct and incorrect detections. (a) Correct detections. (b) Correct detections with the normalized error larger than 0.1. (c) Incorrect detections.

Moreover, the proposed detector with the glare reduction technique gives better detection rate than the detector without the glare reduction technique, where the detection rates for the images with glasses are 95.6% and 93.5%, respectively.

Fig. 9(a) shows some examples of the correct detections. As can be seen in the figure, the proposed detector gives robust performance to several kinds of variations such as facial pose, illumination, and closed eyes. Fig. 9(b) shows some examples of the correct detections with the normalized error larger than 0.1. In this figure, the normalized error of each image from left to right is 0.106, 0.108, and 0.122, respectively. Fig. 9(c) shows some examples of the incorrect detections, where the normalized error of each image from left to right is 0.130, 0.160, and 0.297, respectively. Even though the detection rate for the images with glasses is more than 95% due to the glare reduction technique, incorrect detections are mainly caused by glare on glasses.

On average, the execution time of the proposed detector is 5.5 ms on an Intel i7 3.2-GHz CPU.

C. Test Results for the Real-World Data Set

In the previous section, we used the CMU PIE database to evaluate the proposed detector under several kinds of variations such as facial pose, illumination, glasses, and closed eyes. The facial images in the CMU PIE database are obtained in well-controlled environments, i.e., the CMU 3-D room [33]. In this section, a real-world data set is used to test the proposed detector. We obtained facial images from the web and personal digital cameras. The data set contains 2331 facial images obtained from 281 subjects, in which there are various ethnicities such as Caucasian, Black, East Asian, and Indian, both men and women and both old and young. Those images have pose variations up to 45° in all directions, including roll, yaw, and pitch, and have large illumination variations. Fig. 10 shows some examples of eye detection results for the data set, selecting only persons who agreed to publish their face images. The proposed detector gives good performance under conditions of pose and illumination variations, glasses, and partial occlusion by hair.

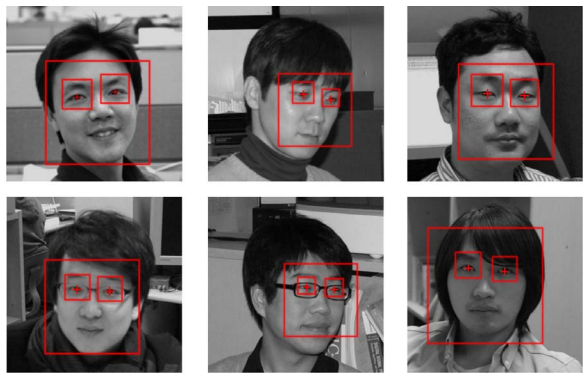


Fig. 10. Examples of eye detection results for the real-world data set.

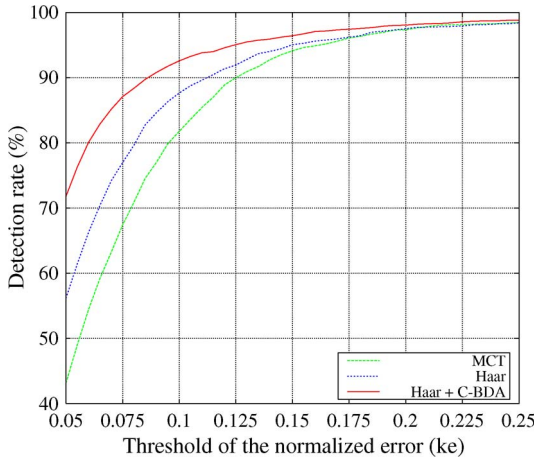


Fig. 11. Detection rates with respect to the threshold of the normalized error for the real-world data set.

We compared the performance of the proposed detector with those of other detectors using MCT features [16] or Haar-like features [22]. Fig. 11 shows the eye detection results with respect to the threshold of the normalized error for the real-world data set. As shown in Fig. 11, the proposed detector using both Haar-like and C-BDA features gives better performance than the detectors using MCT features or Haar-like features. If the threshold is 0.1, the proposed detector gives a detection rate of 92.6%, which is 5.0% and 10.8% higher than that of the detectors using Haar-like features and MCT features, respectively. This means that C-BDA features enable more accurate eye detection than Haar-like features and MCT features for the real-world images. The proposed detector gives detection rates of 95.1%, 96.4%, and 98.1%, when k_e is 0.125, 0.15, and 0.20, respectively.

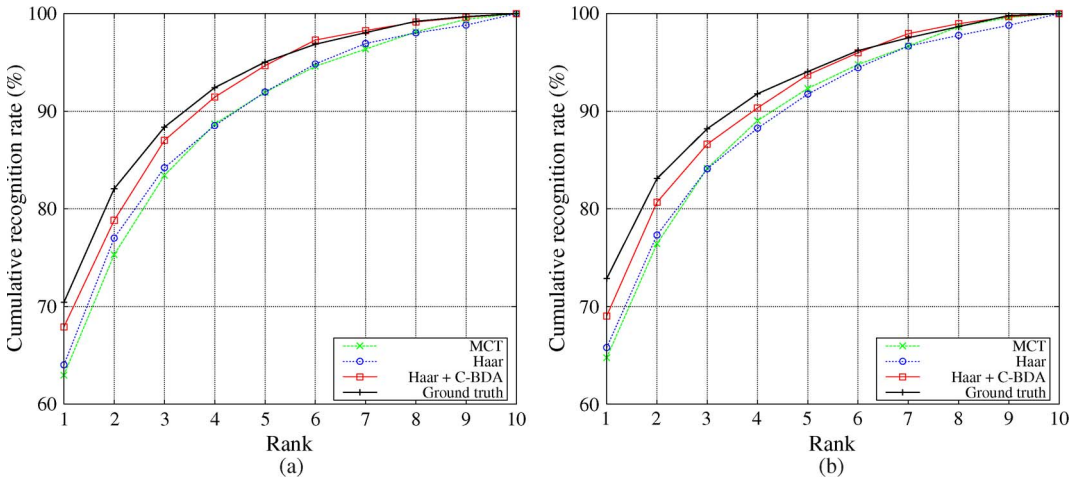


Fig. 12. Face recognition results for the real-world data set: Ten subjects are randomly chosen, and a randomly chosen image of each subject is used for target image of the gallery; the remaining images of each subject are used for query images of the probe. (a) PCA+LDA. (b) C-LDA.

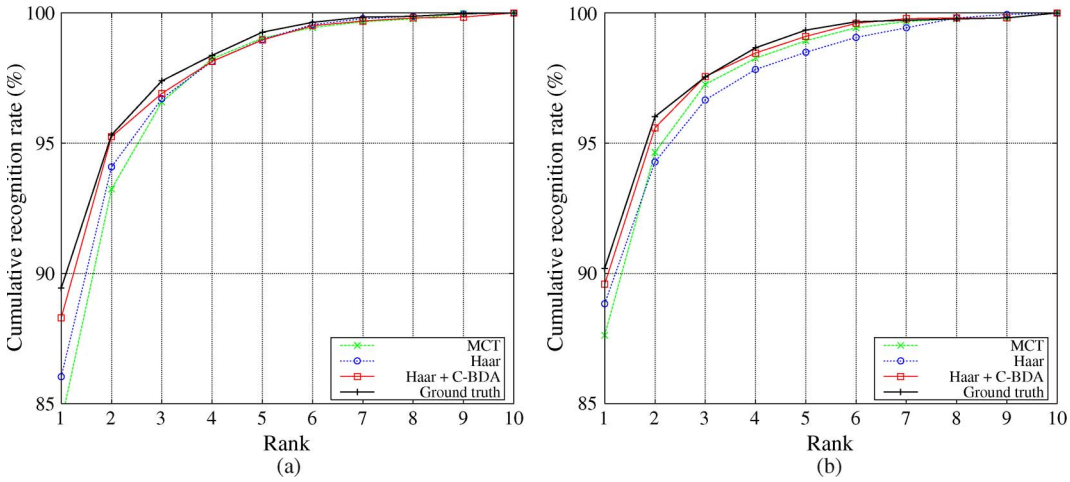


Fig. 13. Face recognition results for the real-world data set: Ten subjects are randomly chosen, and the leave-one-out method is used for test. (a) PCA+LDA. (b) C-LDA.

As mentioned in the introduction, the accuracy of the eye coordinates greatly affects the performance of a face recognition system. We investigated the face recognition performance using the eye coordinates from the proposed detector and compared its performance with those of other eye detectors. For the face recognition experiments, 1280 images from the CMU PIE and BioID databases are used for training, and 651 images of 100 subjects in the real-world data set are used for testing.

After obtaining the eye coordinates from an eye detector, the eyes are aligned horizontally by rotation. Each face is cropped in proportion to the interocular distance and is rescaled to a size of 100×120 (width $\times$ height) as in [9]. Then, histogram equalization is applied to the rescaled image for illumination normalization. The features for face recognition are extracted from these 100×120 normalized face images by using PCA+LDA [41], [42] or C-LDA [35].

The face recognition performance was evaluated by using the cumulative match score [43]. For each round, ten subjects are randomly chosen in the test set, and a randomly chosen image of each subject is used for target image t_k $(k = 1, \ldots, 10)$ of the gallery, while the remaining images of each subject are used for query images of the probe. For each query image q_i, we compute the similarity measure $s_i(k)$ for each image t_k in the gallery. For a given query image q_i, the target images t_k's are sorted by the similarity scores $s_i(\cdot)$. The Euclidean distance was used to calculate the similarity scores. In order to ensure the reliability of the results, we repeated the aforementioned process 50 times and reported the average results. Fig. 12 shows the experimental results, where the horizontal axis corresponds to the rank k and the vertical axis corresponds to the cumulative recognition rate of the top k matches. In the figure, the ground truth means that each face image is aligned using manually located eye coordinates. As shown in the figure, the proposed C-BDA detector shows similar performance to the ground truth. In the case of PCA+LDA, the difference is 2.5%, 1.4%, and 0.4%, when k is 1, 3, and 5, respectively. The proposed detector shows better performance than the methods using Haar-like features and MCT features. The tendency is similar in the case of C-LDA. On the whole, C-LDA shows slightly better performance than PCA+LDA.

In the previous face recognition experiments, only a single image for each subject is used for the gallery, and therefore,

the recognition rates are not high. In the next experiments, ten subjects are randomly chosen, and then, the leave-one-out method is used for testing. In the leave-one-out method, only a single image is used for a query image, and the remaining images are used for the gallery. This is repeated such that each image from ten subjects is used once as the query image. We also repeated the aforementioned process 50 times, and in each round, there are face images of randomly chosen ten subjects. Fig. 13 shows the experimental results. On the whole, face recognition rates in Fig. 13(a) and (b) are better than those in Fig. 12(a) and (b). As in the experiments in Fig. 12, the proposed detector shows similar performance to the ground truth. In the case of C-LDA, the difference is 0.6%, 0.4%, and 0.0%, when k is 1, 2, and 3, respectively. The proposed detector shows better performance than the methods using Haar-like features and MCT features. From these face recognition experiments, we can see that the accuracy of the proposed eye detector is comparable with that of the ground-truth data in the context of their intended application.

IV. Conclusion

In this paper, we proposed a new method called C-BDA for eye detection, which is a critical first step in face verification and recognition systems. The proposed C-BDA has several advantages over BDA. First, C-BDA is a general method using the covariance of composite vectors instead of the covariance of pixels and can be considered as a generalization of BDA. Next, considering the composite feature in C-BDA is obtained by a linear combination of composite vectors, i.e., a linear combination of rectangular windows, it preserves locality better than the primitive feature in BDA does. Third, C-BDA performed better than BDA and other methods as described in Section III. Particularly in the hybrid cascade detector constructed for eye detection, the composite features removed false positives more efficiently in the later stages. The experimental results for the CMU PIE database and the real-world data set showed that the proposed detector provides robust performance to several kinds of variations and much more accurate detection than other methods. Also, in the face recognition experiments, the accuracy of the proposed eye detector showed comparable performance with that of the ground-truth data. These results indicate that the covariance of composite vectors captures the discriminative information better than the covariance of pixels does.

References

[1] T. D'Orazio, M. Leo, G. Cicirelli, and A. Distante, "An algorithm for real time eye detection in face images," in *Proc. Int. Conf. Pattern Recognit.*, 2004, vol. 3, pp. 278–281.

[2] J. Huang and H. Wechsler, "Eye detection using optimal wavelet packets and radial basis functions (RBFs)," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 13, no. 7, pp. 1009–1025, 1999.

[3] Y. Ma, X. Ding, Z. Wang, and N. Wang, "Robust precise eye location under probabilistic framework," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2004, pp. 339–344.

[4] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 84–91.

[5] J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information," *Pattern Recognit.*, vol. 39, no. 6, pp. 1110–1125, Jun. 2006.

[6] P. Wang and Q. Ji, "Learning discriminant features for multi-view face and eye detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 373–379.

[7] P. Wang and Q. Ji, "Multi-view face and eye detection using discriminant features," *Comput. Vis. Image Understanding*, vol. 105, no. 2, pp. 99–111, Feb. 2007.

[8] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Comput. Vis. Image Understanding*, vol. 98, no. 1, pp. 124–154, Apr. 2005.

[9] C. Kim and C.-H. Choi, "Image covariance-based subspace method for face recognition," *Pattern Recognit.*, vol. 40, no. 5, pp. 1592–1604, May 2007.

[10] P. Wang, M. B. Green, Q. Ji, and J. Wayman, "Automatic eye detection and its validation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 3, p. 164.

[11] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[12] T. Kawaguchi and M. Rizon, "Iris detection using intensity and edge information," *Pattern Recognit.*, vol. 36, no. 2, pp. 549–562, 2003.

[13] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[14] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 748–763, Jun. 2002.

[15] C. Kim, M. Turk, and C.-H. Choi, "Biased discriminant analysis using composite vectors for eye detection," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2008, pp. 1–6.

[16] I. Choi and D. Kim, "Eye correction using correlation information," in *Proc. Asian Conf. Comput. Vis.*, 2007, pp. 698–707.

[17] M. Song, D. Tao, Z. Sun, and X. Li, "Visual-context boosting for eye detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1460–1467, Dec. 2010.

[18] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 32–39.

[19] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. 366–373.

[20] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. Int. Conf. Image Process.*, 2002, vol. 1, pp. 900–903.

[21] Y. F. Pan, X. Hou, and C. L. Liu, "A robust system to detect and localize texts in natural scene images," in *Proc. IAPR Workshop Doc. Anal. Syst.*, 2008, pp. 35–42.

[22] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[23] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 555–562.

[24] C. Kim and C.-H. Choi, "A discriminant analysis using composite features for classification problems," *Pattern Recognit.*, vol. 40, no. 11, pp. 2958–2966, Nov. 2007.

[25] X. S. Zhou and T. S. Huang, "Small sample learning during multimedia retrieval using BiasMap," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. 11–17.

[26] T. S. Huang, X. S. Zhou, M. Nakazato, Y. Wu, and I. Cohen, "Learning in content-based image retrieval," in *Proc. Int. Conf. Develop. Learn.*, 2002, pp. 155–162.

[27] M. Nakazato, C. Dagli, and T. S. Huang, "Evaluating group-based relevance feedback for content-based image retrieval," in *Proc. Int. Conf. Image Process.*, 2003, vol. 2, pp. 599–602.

[28] D. Tao, X. Tang, X. Li, and Y. Rui, "Direct kernel biased discriminant analysis: A new content-based image retrieval relevance feedback algorithm," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 716–727, Aug. 2006.

[29] L. Wang, K. L. Chan, and P. Xue, "A criterion for optimizing kernel parameters in KBDA for image retrieval," *IEEE Trans. Syst., Man, Cybern. B*, vol. 35, no. 3, pp. 556–562, Jun. 2005.

[30] W. Bian and D. Tao, "Biased discriminant Euclidean embedding for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 545–554, Feb. 2010.

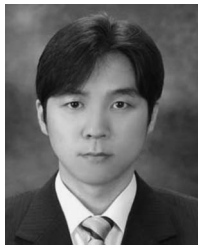
[31] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.

[32] D. Xu, S. Yan, D. Tao, S. Lin, and H.-J. Zhang, "Marginal Fisher analysis and its variants for human gait recognition and content-based image

retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2811–2821, Nov. 2007.
[33] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 46–51.
[34] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[35] C. Kim, "Pattern recognition using composite features," Ph.D. dissertation, Seoul Nat. Univ., Seoul, Korea, 2007.
[36] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the Hausdorff distance," in *Proc. Int. Conf. Audio- Video-Based Biometric Person Authentication*, 2001, pp. 90–95.
[37] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2004, pp. 91–96.
[38] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Comput.*, vol. 11, pp. 1691–1715, 1999.
[39] F. Fleuret and D. Geman, "Coase-to-fine face detection," *Int'l Journal of Computer Vision*, vol. 41, pp. 85–107, 2001.
[40] J. Quinlan, "Introduction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, 1986.
[41] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 831–836, Aug. 1996.
[42] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[43] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

Chunghoon Kim (M'11) received the B.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 1998 and 2007, respectively.

He is currently a Staff Engineer with the Qualcomm Research, Seoul, Korea. Prior to joining Qualcomm, he was a Senior Engineer with Samsung Electronics from 2009 to 2010 and a Postdoctoral Researcher with the University of California, Santa Barbara, from 2007 to 2009. His research interests include pattern recognition, machine learning, computer vision, and their applications.



Sang-Il Choi (S'05–M'10) received the B.S. degree in the division of electronic engineering from Sogang University, Seoul, Korea, in 2005 and the Ph.D. degree from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, in 2010.

He was a Postdoctoral Researcher in the BK21 Information Technology, Seoul National University, in 2010 and in the Institute for Robotics and Intelligent Systems of Computer Science Department, University of Southern California, Los Angeles, until August of 2011. He is currently an Assistant Professor with the Department of Applied Computer Engineering, Dankook University, Gyeonggi-do, Korea. His research interests include pattern recognition, feature extraction and selection, machine learning, computer vision, and their applications.

Matthew Turk (S'87–M'91–SM'99) received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1991.

He is currently a Professor of Computer Science and Media Arts and Technology with the University of California, Santa Barbara, where he codirects the UCSB Four Eyes Lab, with a research focus on the "four I's" of Imaging, Interaction, and Innovative Interfaces. He has worked at Martin Marietta Denver Aerospace, LIFIA/ENSIMAG (in Grenoble, France), Teleos Research, and at Microsoft Research, where he was a founding member of the Vision Technology Group. He is a founding member and former Chair of the advisory board for the International Conference on Multimodal Interfaces and on the editorial board of the Journal of Image and Vision Computing and the ACM Transactions on Intelligent Interactive Systems.

Dr. Turk was a General Chair of the recent 2011 IEEE Conference on Automatic Face and Gesture Recognition and will be the General Chair of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. He received the 2011–2012 Fulbright-Nokia Distinguished Chair in Information and Communications Technologies.

Chong-Ho Choi (S'77–M'78) received the B.S. degree from Seoul National University, Seoul, Korea, in 1970 and the M.S. and Ph.D. degrees from the University of Florida, Gainesville, in 1975 and 1978, respectively.

He was a Senior Researcher with the Korea Institute of Science and Technology from 1978 to 1980. He is currently a Professor with the School of Electrical Engineering and Computer Science, Seoul National University. He is also affiliated with the Automation and Systems Research Institute, Seoul National University. His research interests include control theory and network control, neural networks, system identification, and their applications.