Biased Discriminant Analysis Using Composite Vectors for Eye Detection

Chunghoon Kim, Matthew Turk
Computer Science Department
University of California, Santa Barbara
{chkim, mturk}@cs.ucsb.edu

Chong-Ho Choi
Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea
chchoi@csl.snu.ac.kr

Abstract

We propose a new discriminant analysis using composite vectors for eye detection. A composite vector consists of a number of pixels inside a window on an image. The covariance of composite vectors is obtained from their inner product and can be considered as a generalized form of the covariance of pixels. The proposed C-BDA is a biased discriminant analysis using the covariance of composite vectors. In the hybrid cascade detector constructed for eye detection, Haar-like features are used in the earlier stages and composite features obtained from C-BDA are used in the later stages. The experimental results for the CMU and Yale databases show that the proposed detector provides robust performance to several kinds of variations such as facial pose, illumination, and closed eyes. In particular, it provides a 99.4% detection rate for the CMU images without glasses.

1. Introduction

Recently, several studies have been done on eye detection as a preprocessing step for face recognition [3, 10, 11, 13, 15, 16]. After detecting faces in an image, it is necessary to align faces for face recognition. Face alignment is usually performed by using the coordinates of the left and right eyes, and the accuracy of the eye coordinates affects the performance of a face recognition system [7, 13, 15]. According to recent results in the field of face recognition, state-of-the-art methods provide a recognition rate reaching almost 100% even under variations in facial expression and illumination [7, 9]. In those experiments, the eye coordinates were manually located. When these coordinates were shifted randomly, the recognition rates degraded rapidly [7, 15]. From these results, we can see that eye detection is very important in face recognition systems.

In the previous studies, several kinds of features were used to discriminate between eyes and non-eyes. Pentland *et al*. used the Eigeneyes based on principal component analysis (PCA) [11]. Huang and Wechsler used wavelet packets for eye representation and radial basis functions for classification of eyes and non-eyes [3]. Ma *et al*. used Haar-like features to find the possible eyes [10]. Wang and Ji used features obtained from the recursive nonparametric discriminant analysis to find the face and eyes [15, 16].

On the other hand, Kim and Choi introduced a new method of extracting composite features for classification problems [6, 7]. In their study, a composite vector is composed 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 technique (C-LDA) is derived by using the covariance of composite vectors. In C-LDA, features are obtained by linear combinations of the composite vectors and 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, C-LDA showed good performance when adjacent primitive variables are strongly correlated as in image data sets and the Sonar data set [6, 7].

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) [18]. 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 the normal distribution of negative samples and is an appropriate method in detection problems.

In this paper, we propose a new biased discriminant analysis using composite vectors, called C-BDA, for eye detection. C-BDA is derived from biased discriminant analysis by using the covariance of composite vectors instead of the covariance of pixels. When detecting 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 majority of

978-1-4244-2154-1/08/$25.00

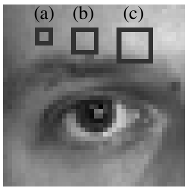


Figure 1. Three sizes of windows, each of which makes a composite vector: The window sizes of (a), (b), and (c) are 4×4, 6×6, and 8×8, respectively.

non-eyes [8, 14]. At the later stages, composite features obtained from C-BDA are used to discriminate between eyes and non-eyes, which are difficult to discriminate by Haar-like features. The experimental results for the CMU [12] and Yale [1] databases show that the proposed detector provides robust performance to several kinds of variations such as facial pose, illumination, and closed eyes.

2. Biased discriminant analysis 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 biased discriminant analysis using the covariance of composite vectors. Unlike [6], we differentiate the composite feature from the composite vector in order to avoid confusion in terminology.

2.1. Composite vectors and their covariance

In general, a pattern is represented by a set of variables, which are called primitive variables [6]. In appearance-based models, the intensity of each pixel in an image is used as a primitive variable. Traditional methods such as PCA and LDA use the covariance of primitive variables, which contains second-order statistical information. Now, let us consider a composite vector composed of a number of primitive variables which correspond to pixels inside a window on an image.

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 $\frac{p}{l}$ where p is the total number of pixels in A. Obviously, more windows can be obtained if neighboring windows overlap each other. Figure 1 shows three sizes of windows. In the figure, the size of the image is 40×40 (pixels), and the window sizes of (a), (b), and (c) are 4×4, 6×6, and 8×8, respectively.

Let the set of composite vectors X be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_1 = \mathcal{O}_\mathcal{L}(H_1)$, $\mathbf{x}_2 = \mathcal{O}_\mathcal{L}(H_2)$, and so on. Here, $\mathcal{O}_\mathcal{L}(\cdot)$ is the lexicographic ordering operator that transforms a matrix into a vector by ordering the rows of the matrix one after the other. Therefore, $\mathbf{x}_i$ becomes an l-dimensional vector. For a sample set of N images, let C denote a covariance matrix based on the composite vectors. The element c_{ij} of C is defined as [6]

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

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 dependency among multiple pixels. When l is 1, $\mathbf{x}_i$ becomes a scalar and c_{ij} in (1) is the same as the covariance of primitive variables. Therefore, the covariance of composite vectors can be considered as a generalized form of the covariance of primitive variables.

Then, the covariance matrix C is computed as [17]

$$C = \frac{1}{N}\sum_{k=1}^{N}(X(k) - M)(X(k) - M)^T, \quad (2)$$

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 $C \in \mathbb{R}^{n \times n}$.

2.2. Analysis of the covariance matrix based on composite vectors

Let us investigate the covariance matrix C in (2). 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 (2) as

$$C = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{l}(\boldsymbol{\chi}_j(k) - \mathbf{m}_j)(\boldsymbol{\chi}_j(k) - \mathbf{m}_j)^T. \quad (3)$$

The number of outer products of vectors in (3) is Nl, which is l times larger than that in a covariance matrix based on primitive variables [17]. This is because $X(k)$ has l column vectors of $\boldsymbol{\chi}_j(k)$. Figure 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, respectively, because the 4×4 windows are used for making the composite vectors and they overlap either horizontally or vertically by 50%. For visualization, $\boldsymbol{\chi}_j(k)$'s are represented as images. As can be seen in the figure, there is a small variation in eye positions of $\boldsymbol{\chi}_j(k)$ images. All these 16 images are used for making C in (3) as if they are individual samples. If l becomes larger, more images with a larger variation are used for making C.

Let us further investigate C in (3). We rewrite (3) as

$$C = \sum_{j=1}^{l} S_j, \quad (4)$$

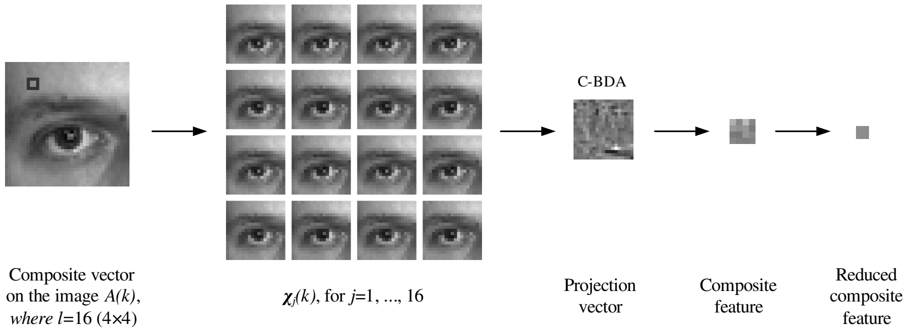


Figure 2. Schematic diagram of C-BDA: In this case, the size of $A(k)$ is 40×40, the size of the composite vector l is 16, and the number of composite vectors n is 361. For visualization, $\boldsymbol{\chi}_j(k)$'s are represented as images. The projection vector is obtained by C-BDA and is represented as a 19×19 image in the figure. Note that the composite feature has the same size as the composite vector and is further reduced by applying a downscaling operator. In this case, the downscaling factor r is 16.

where $S_j = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{\chi}_j(k) - \mathbf{m}_j)(\boldsymbol{\chi}_j(k) - \mathbf{m}_j)^T$. Here, S_j is the primitive covariance matrix which corresponds to the covariance matrix based on primitive variables. Since 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 (1), where the covariance c_{ij} is defined as the sum of l covariances between the corresponding pixels in $\mathbf{x}_i$ and $\mathbf{x}_j$.

2.3. BDA using the covariance of composite vectors (C-BDA)

The composite vectors and their covariance are obtained in Section 2.1. Now, let us derive a new discriminant analysis using the covariance of composite vectors 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 biased discriminant analysis (BDA) [18]. 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 the normal distribution of negative samples and is an appropriate method in detection problems.

Let us now derive C-BDA, which is a biased discriminant analysis 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 $X_P(k)$ and $X_N(k)$ denote the sets of composite vectors of the kth positive sample and the kth negative sample, respectively. Here, $X_P(k), X_N(k) \in \mathbb{R}^{n\times l}$. In C-BDA, the set of projection vectors W_B is obtained by

$$W_B = \arg\max_{W} \frac{|W^T C_N W|}{|W^T C_P W|}, \qquad (5)$$

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

$$C_P = \frac{1}{N_P}\sum_{k=1}^{N_P}(X_P(k) - M_P)(X_P(k) - M_P)^T, \qquad (6)$$

$$C_N = \frac{1}{N_N}\sum_{k=1}^{N_N}(X_N(k) - M_P)(X_N(k) - M_P)^T. \qquad (7)$$

Here, $M_P = \frac{1}{N_P}\sum_{k=1}^{N_P} X_P(k)$ is the mean of the positive samples, and N_P and N_N are the number of positive and negative samples, respectively. The optimization problem of (5) can be computed in two steps as in C-LDA [6]. After whitening the C_P, C-BDA finds a linear transform by which negative samples are as far away from the mean of the positive samples as possible.

Then, the set of features $Y(k)$ is obtained from $X(k)$ as

$$Y(k) = {W_B}^T X(k), \qquad k = 1, 2, \ldots, N, \qquad (8)$$

where $Y(k) \in \mathbb{R}^{m\times l}$ has m features $[\mathbf{y}_1(k) \ \mathbf{y}_2(k) \ \ldots \ \mathbf{y}_m(k)]^T$. 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 $\mathbf{x}_n(k)$. It is noted that C-BDA with $l = 1$ is the same as BDA.

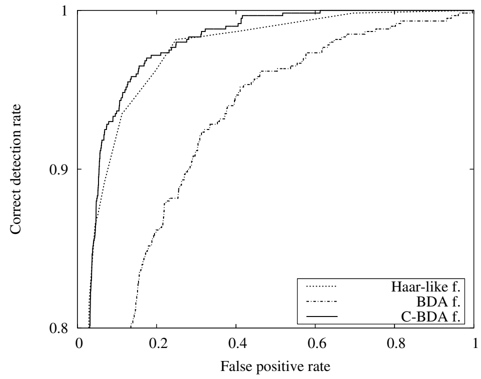
The first projection vector $\mathbf{w}_1$ is represented in Fig. 2. In this case, n is 361 and $\boldsymbol{\chi}_j(k)$'s are 361-dimensional vectors. Therefore, $\mathbf{w}_1$ is a 361-dimensional vector and is represented as a 19×19 image in the figure. The composite feature in the figure is obtained by projecting $\boldsymbol{\chi}_j(k)$'s onto the projection vector. Thus, the composite feature is a 16-dimensional 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 [7]. Therefore, the dimension of the composite feature can be reduced significantly. In this case, the downscaling factor r

(a) Positive samples

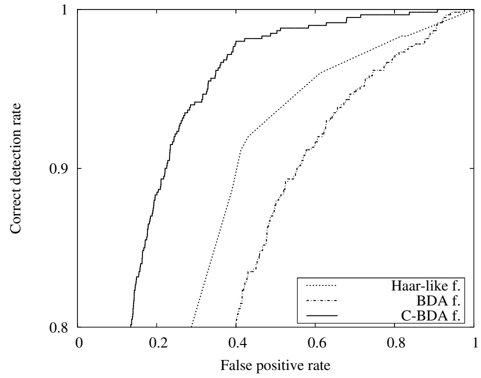
(b) Negative samples at the first stage

(c) Negative samples at the fifth stage

Figure 3. Eye and non-eye samples used for training.



(a) First stage



(b) Fifth stage

Figure 4. ROC curves comparing C-BDA features with Haar-like features at the first and fifth stages of the cascade detector.

is set to 16, so the 4×4 elements of each composite feature are represented by their average value.

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

3. Experimental results

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

3.1. Training of the hybrid cascade detector

For training the detector, 1,800 images are collected from the CMU PIE database [12], the BioID database [4], and our digital camera images, which include several kinds of variations such as glasses, pose, illumination, closed eyes, partial occlusion, and low resolution. From these images, faces are cropped by a face detector using the modified census transform [2], and are rescaled to a size of 80×80 (pixels). Based on the symmetry of the face, only a right eye detector is trained as in [15], 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 images. Some positive and negative samples are shown in Fig. 3(a) and Fig. 3(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 0.8 times the interocular distance, and is rescaled to a size of 18×18 (pixels).

When detecting the eye coordinates in an 80×80 face image, four sizes of detection windows, 18×18, 22×22, 26×26, and 31×31, are scanned across the image [14]. A large number of detection windows are used for detection and the majority of detection windows are non-eyes. In this case, a cascade of classifiers is an efficient way to detect eye coordinates [8, 14]. 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.

Haar-like features and composite features can be used for classifiers to discriminate between eyes and non-eyes. Haar-like features are easy to compute [14], while composite features have a powerful discriminative information. 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 non-eyes. At the later stages, composite features obtained from C-BDA are used to discriminate between eyes and non-eyes, which are difficult to discriminate by Haar-like features.

Figure 4(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, 1,200 images randomly chosen from the 1,800 images are used for training and the remaining 600 images are used for validation. In the training set, there are 1,200 positive samples obtained from right eyes and 1,200 negative samples obtained from the other regions of images. In the validation set, there are 600 positive and 6,000 negative samples, where more negative samples are used for making the validation phase as a real detection situation. 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 four features of each method. As can be seen in Fig. 4(a), when the false positive rate is 40%, the detection rates of Haar-like, BDA, and C-BDA features are 98.7%, 94.7%, and 99.0%, respectively. 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 4, 4, 14, and 18 Haar-like features, respectively.

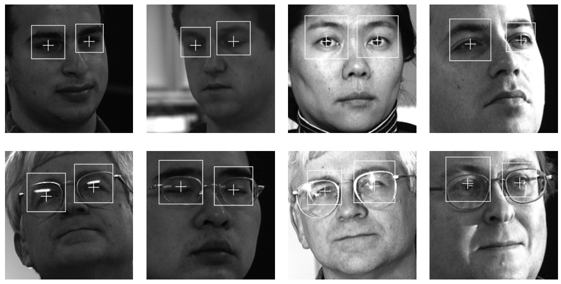
Although the Haar-like features are easy to compute, they have limited discriminative power [10, 16]. Especially at the later stages in the cascade detector, the Haar-like features can not remove false positives efficiently, while correctly detecting eyes. Some negative samples at the fifth stage are shown in Fig. 3(c), where they are obtained from false positives in the previous cascade detector [14]. As can be seen in the figure, most of negative samples are obtained near the eye position of face images. Figure 4(b) shows the ROC curves on the validation set of the fifth stage. In C-BDA, the 2×2 windows are used for making the composite vectors in an 18×18 image and the downscaling factor r is set to 4. When the false positive rate is 40%, the C-BDA features provide a detection rate of 98.0%, which is 8.3% and 18.0% 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 4, 6, 12, and 16 C-BDA features, respectively.

3.2. Test results for the CMU database

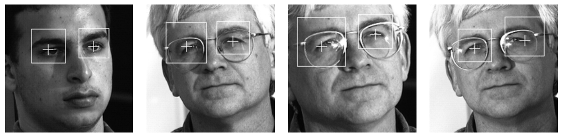
The CMU PIE database is used to evaluate the robustness of the proposed detector to glasses, pose, illumination, and closed eyes [12]. In this experiment, we used 3,604 face images of 68 people, which were not used in the training. Among 68 people, 28 people are wearing glasses and 40 people are not. There are 8 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. There are 24 images per person under different illumination conditions, named from '27(05)_00' to '27(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.

Table 1. Eye detection results for the CMU PIE database

variation	w/o glasses	with glasses	average
pose (open)	100%	91.1%	96.3%
pose (closed)	99.4%	88.4%	94.9%
illum. (frontal)	99.6%	94.7%	97.6%
illum. (3/4 profile)	99.2%	93.8%	96.9%
total	99.4%	93.5%	97.0%



(a) correct detections



(b) incorrect detections

Figure 5. Examples of the correct and incorrect detections.

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 distance between manually and automatically located coordinates of the left and right eye, respectively. Then, the normalized error e_n is computed as

$$e_n = \frac{\max(e_l, e_r)}{d_{lr}}. \tag{9}$$

We consider a detection result as correct if $e_n \leq k_e$ and typically set $k_e = 0.125$ as in [13].

Table 1 shows the detection results for the CMU PIE database. 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 99.4% detection rate for the images without glasses. Figure 5(a) shows some examples of the correct detections. When k_e is 0.125, 0.15, and 0.20, the total detection rates are 97.0%, 98.1%, and 99.3%, respectively. Incorrect detections are mainly caused by glare on glasses. Figure 5(b) shows some examples of the incorrect detections. The normalized error of each image from left to right is 0.129, 0.141, 0.167, and 0.212, respectively.

On average, the execution time of the proposed detector is 27.9 (msec) on a Core2Duo 2.0GHz CPU.

3.3. Test results for the Yale database

The Yale database is used to evaluate the accuracy of the proposed detector and face recognition performance [1]. The Yale database contains 165 gray images of 15 individuals, gathered with different facial expressions, with or without glasses, and under different lighting conditions. When k_e is 0.125 and 0.15, the detection rates for 165 images are 96.4% and 98.8%, respectively. The average of both e_l and e_r is 2.1 pixels in 243×320 resolution, which is better than the result in [5].

After detecting eyes by using the proposed 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 120×100 as in [7]. The 11-fold cross validation [17] is used to evaluate the face recognition performance, where C-LDA [7] gives 100% and 97.6% recognition rates when faces are manually and automatically aligned, respectively. Incorrect recognitions are mainly caused by the false eye detections. When k_e=0.125, 99.4% of correctly detected images are correctly recognized in face recognition. When k_e=0.15, 98.8% of correctly detected images are correctly recognized, and two incorrectly detected images are incorrectly recognized. From this result, we can see that correctly detected images are expected to be correctly recognized in face recognition.

4. Conclusions

In this paper, we proposed C-BDA for eye detection. C-BDA is a general method using the covariance of composite vectors instead of the covariance of primitive variables, and it can be applied to other detection problems as well. In the hybrid cascade detector constructed for eye detection, composite features obtained from C-BDA were used in the later stages to remove false positives efficiently. The experimental results for the CMU and Yale databases showed that the proposed detector provides robust performance to several kinds of variations.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2007-357-D00161). Portions of the research in this paper use the face detector developed by Olaworks (http://www.olaworks.com).

References

[1] The Yale Face Database. http://cvc.yale.edu/projects/ yalefaces/yalefaces.html.

[2] B. Froba and A. Ernst. Face detection with the modified census transform. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 91–96, 2004.

[3] J. Huang and H. Wechsler. Eye detection using optimal wavelet packets and radial basis functions (RBFs). *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 13(7):1009–1026, 1999.

[4] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the Hausdorff distance. In *Proc. Int'l Conf. Audio- and Video-based Biometric Person Authentication*, pages 90–95, 2001.

[5] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia. A hybrid classifier for precise and robust eye detection. In *Proc. Int'l Conf. Pattern Recognition*, volume 4, pages 731–735, 2006.

[6] C. Kim and C.-H. Choi. A discriminant analysis using composite features for classification problems. *Pattern Recognition*, 40:2958–2966, 2007.

[7] C. Kim and C.-H. Choi. Image covariance-based subspace method for face recognition. *Pattern Recognition*, 40:1592–1604, 2007.

[8] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. Int'l Conf. Image Processing*, volume 1, pages 900–903, 2002.

[9] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Trans. Image Processing*, 11(4):467–476, 2002.

[10] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 339–344, 2004.

[11] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[12] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 46–51, 2002.

[13] J. Song, Z. Chi, and J. Liu. A robust eye detection method using combined binary edge and intensity information. *Pattern Recognition*, 39:1110–1125, 2006.

[14] P. Viola and M. J. Jones. Robust real-time face detection. *Int'l Journal of Computer Vision*, 57(2):137–154, 2004.

[15] P. Wang, M. B. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 3, pages 164–164, 2005.

[16] P. Wang and Q. Ji. Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding*, 105:99–111, 2007.

[17] A. Webb. *Statistical Pattern Recognition*. Wiley, 2nd edition, 2002.

[18] X. S. Zhou and T. S. Huang. Small sample learning during multimedia retrieval using BiasMap. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 11–17, 2001.