Discriminative Feature Co-occurrence Selection for Object Detection

Takeshi Mita, *Member, IEEE*, Toshimitsu Kaneko,
Björn Stenger, *Member, IEEE,* and Osamu Hori, *Member, IEEE*

***Abstract*— This paper describes an object detection framework that learns the discriminative co-occurrence of multiple features. Feature co-occurrences are automatically found by Sequential Forward Selection at each stage of the boosting process. The selected feature co-occurrences are capable of extracting structural similarities of target objects leading to better performance. The proposed method is a generalization of the framework proposed by Viola and Jones, where each weak classifier depends only on a single feature. Experimental results obtained using four object detectors, for finding faces and three different hand poses, respectively, show that detectors trained with the proposed algorithm yield consistently higher detection rates than those based on their framework while using the same number of features.**

***Index Terms*— Object detection, Co-occurrence, Boosting, Feature selection.**

I. INTRODUCTION

THIS paper presents a new method for constructing accurate and efficient classifiers for detecting objects in images. Effective object detection techniques have been developed in recent years that solve a two-class problem by using a probabilistic framework or by finding a discriminant function from a large set of training samples. For example, neural network-based face detectors were used in [1], [2], classification boundaries were learned by support vector machines in [3], [4], [5] and a statistical method based on the naïve Bayes classifier was proposed in [6]. Some of these methods use raw pixel values as features. However, they are sensitive to the addition of noise and change in illumination. Haar wavelet features [5] and rectangle features [7], which extract local gradient, and Gabor features, which respond to particular directions and spatial frequencies of texture, increase the robustness against such effects. Viola and Jones [7] proposed a framework for selecting discriminative features and training classifiers using AdaBoost [13], showing that good classification performance can be achieved by a small number of selected features at much reduced computational cost. Their approach, which selects only discriminative features using boosting from a huge feature pool, has become a general framework for object detection and several extensions have been proposed. We will introduce two of them and then point out problems of these approaches. The first extension is an improvement of the boosting algorithm itself. There are modified versions of AdaBoost such as Real AdaBoost [8], KLBoosting [10] and FloatBoost [11]. FloatBoost was applied not only to face detection but also to hand detection [24]. The second extension is to use an extended set of features such that various image patterns can be evaluated. As shown in Figure 1, in addition to the basic feature set (a), different arrangements or numbers of rectangles such as (b) to (d) are used in [7], [9]. Lienhart et al. [12] introduced an efficient scheme for calculating features rotated by 45°. Kölsch and Turk [25] added new types of rectangle features for robust hand detection. Viola et al. incorporated motion information in addition to appearance into pedestrian detection [26]. Although both extensions are effective, they are insufficient to achieve more accurate object detection. Most of these methods construct a weak classifier by selecting only one feature from the given feature pool. However, the generalization performance is no longer improved at later stages of the boosting process because the classification task using a single feature becomes more difficult. Viola and Jones reported that features which were selected at later stages yielded error rates between 0.4 and 0.5, whereas features selected in early stages had error rates between 0.1 and 0.3 [7]. Such *'too weak'* classifiers do not contribute to improving the generalization performance, although they can reduce the training error. Any sophisticated boosting algorithm will encounter this problem. However, taking the simplest approach by increasing the variation of rectangle features or including additional feature spaces leads to prohibitively expensive training. It is also difficult to create new discriminative feature sets manually according to each object of interest.

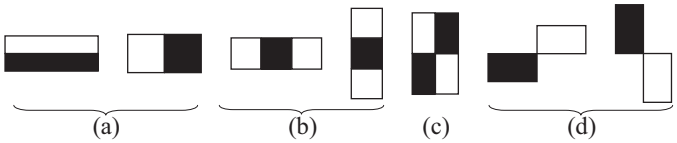

Fig. 1. Examples of rectangle feature sets. Basic feature set (a) consists of only two-rectangle features. Extended feature sets (a) to (d) include features with different numbers and arrangements of rectangles.

In this paper, we propose an object detection framework that achieves higher accuracy at lower computational cost by incorporating the co-occurrence of multiple features at each stage of the boosting process. In the proposed framework, discriminative feature co-occurrences are automatically found by Sequential Forward Selection and weak classifiers based on the best feature co-occurrences are added through the boosting process. Figure 2 compares three methods for learning face detectors based on rectangle features: (a) Viola and Jones' framework, (b) a conventional framework and (c) the proposed framework. The framework (b) constructs a classifier by only selecting a feature co-occurrence without using boosting. In this paper, we regard it as a conventional framework. In our framework (c), the final strong classifier $H(x)$ is a linear combination of weak classifiers $h_1(x)$ to $h_T(x)$. In contrast to the framework (a), each weak classifier of the proposed framework consists of multiple features. For example, $h_1(x)$ observes F features simultaneously and

Manuscript received October 26, 2006; revised June 19, 2007.
T. Mita is with Toshiba Corporation.

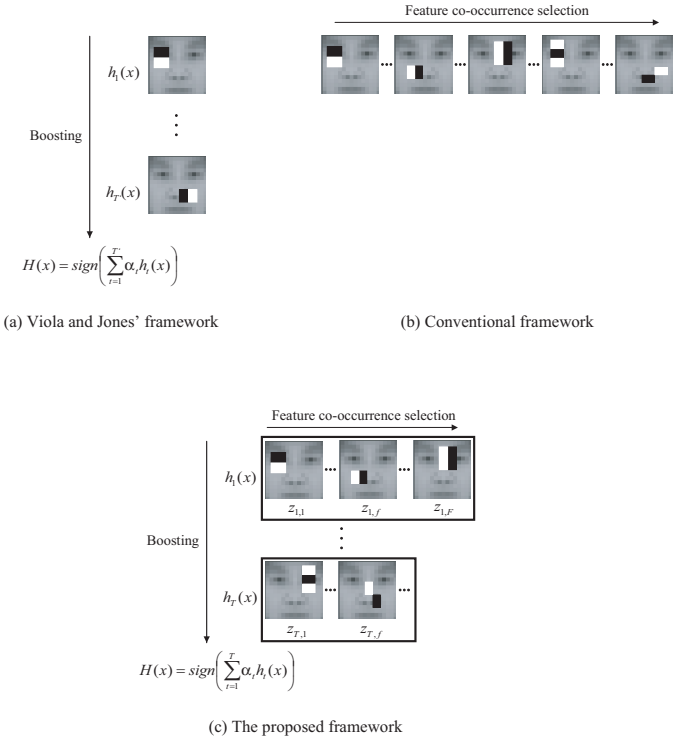

Fig. 2. Comparison of three frameworks: (a) Viola and Jones', (b) a conventional method of constructing a classifier by selecting feature co-occurrence without boosting and (c) the proposed framework. Both (a) and (b) can be regarded as special cases of the proposed framework (c).

evaluates joint statistics of these features. Structural similarities of faces, which cannot be evaluated using a single feature, are extracted from $z_{1,1}$ (eye regions are darker than neighboring regions), $z_{1,f}$ (nostrils are dark) and $z_{1,F}$ (the region between the eyes is brighter than the eyes). These combined features are selected automatically. The combination of spatially separated features as shown in Figure 2 cannot be found by the extended feature set shown in Figure 1. From a different point of view, the proposed framework (c) can be understood as the generalization of (a) and (b) because they are special cases of the proposed framework. Our framework is a solution to the problem of how to choose features in order to build a more accurate classifier without changing the total number of features, i.e. without increasing the computational cost.

In the experiments we compare the performance of several classifiers, each corresponding to a frontal upright face detector and detectors for hands in three different poses. Performance evaluation has been carried out by 10-fold cross validation. An experiment integrating Real AdaBoost [8] into our framework instead of standard AdaBoost [13] shows that further improvement is possible.

Section II shows how co-occurrences of multiple rectangle features are represented. Section III describes an algorithm for constructing an object detector by selecting discriminative feature co-occurrences by Sequential Forward Selection at each stage of the boosting process. Experimental results are shown in Section IV.

II. Feature Co-occurrence Representation

This section describes the representation for co-occurrence of multiple rectangle features. We use rectangle features because they can be computed in constant time at any scale or location.

A. Rectangle Features

Rectangle features have scalar values that represent differences in average intensities between adjacent rectangular regions. They can extract texture information without depending on absolute intensities. They are not too much affected by noise which arises while capturing images or transmission of image signals because they include multiple pixels.

They capture the intensity gradient at different locations, spatial frequencies and directions by changing the position, size, shape and arrangement of rectangular regions exhaustively according to the base resolution of the classifier. For example, when the resolution is 25×25 pixels, 239,408 feature candidates are generated from the feature set (a) to (d) in Figure 1. For each candidate, a one-dimensional probability density is calculated from all training samples as shown in Figure 3. The two curves show the densities from the object and non-object class. In [7], a weak learning algorithm is designed to select the single feature that best separates the classes with a threshold. A small number of discriminative features is selected by updating the sample distribution using AdaBoost. However, as mentioned above, the error rate of the weak classifiers selected at later stages of the boosting process becomes large because the updated sample distribution consists of many difficult samples, which are similar to each other. Even the best feature selected from 239,408 candidates cannot provide good classification performance. Figure 4 shows the performance of Viola and Jones' face detector. The training error (error rate measured on the training samples) and the generalization error (error rate measured on test samples) are plotted against the number of weak classifiers. The training error converges to zero when the number of features reaches about 500. However, the generalization error is no longer reduced after 1,000 features are selected. This means that no discriminative features remain in the pool of candidates and that further improvement cannot be expected. Wu et al. [9] divide the range of the feature values into 64 partitions to increase the classification power. However, the above problem still remains when two class distributions overlap.

B. Feature Value Quantization

To improve the generalization performance we use weak classifiers that include multiple features simultaneously. Feature co-occurrence makes it possible to classify difficult samples that are misclassified by weak classifiers using a single feature. We represent the statistics of feature co-occurrence by using their joint probability. To calculate the joint probability we quantize the feature value z to two levels. By doing so, each feature value is represented by a binary variable s, which is 1 or 0, specifying object or non-object, respectively. The variable s for a sample x is calculated by,

$$s(x) = \begin{cases} 1 & p \cdot z(x) > p \cdot \theta \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where θ is a threshold and p is a parity indicating the direction of the inequality sign. The values of θ and p are determined so that the error rate is minimized. This binarization rule is the same

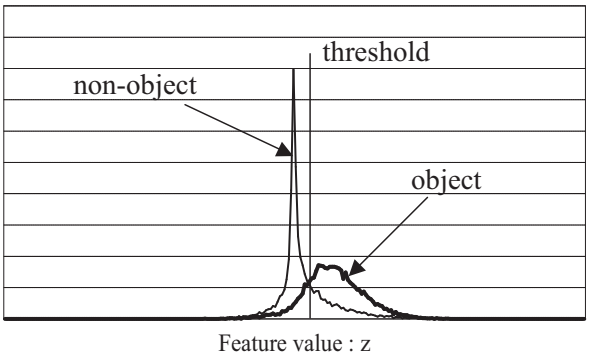

Fig. 3. An example of feature value distributions. A weak learning algorithm is designed to select the single feature that best separates the two classes with a threshold.

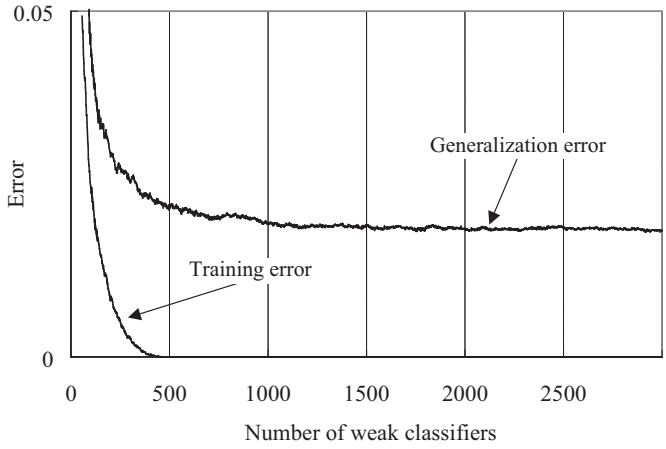

Fig. 4. Performance of Viola and Jones' face detector. The training error converges to zero, but the generalization error is no longer reduced after 1,000 features are selected. This means that no discriminative features remain in the candidate pool and that further improvement cannot be expected. Any sophisticated learning algorithm will encounter this problem if weak classifiers depending on only one feature are used.

as for Viola and Jones' weak classifiers. In order to confirm the effectiveness of exploiting feature co-occurrence, we do not use any operations different from Viola and Jones' framework except for combining multiple features. The proposed framework is not limited to the case of using binarized feature values. Multi-level quantization of the feature value fits more complex distributions than binarization. However, in this paper we do not focus on how many levels are appropriate.

One advantage provided by the binarization is robustness toward image noise and change in illumination. For example, the variable s is invariant to changes in illumination that do not invert the inequality sign in (1).

C. Co-occurrence of Multiple Rectangle Features

The feature co-occurrence is represented by combining the binary variables computed from multiple features. Figure 5 shows an example of the co-occurrence of three rectangle features. When the variables are 1, 0 and 1, the value of the combined features is calculated by

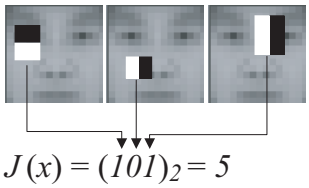

Fig. 5. An example of feature co-occurrence representation. Three binary variables measured from three rectangle features are combined.

$$J(x) = (101)_2 = 5. \tag{2}$$

The value $J(x)$ as a binary number specifies an index for 2^F different combinations, where F is the number of combined features.

For each class statistical dependencies between the features are obtained by observing $J(x_i)$ for each training sample x_i. We use such dependencies for classification. The input pattern is classified to be an object or non-object by evaluating from which class the feature co-occurrence is likely to be observed. The combined features are selected to capture discriminative structural similarities of the samples which belong to the object class. In the next section, we will show the algorithm for selecting discriminative feature co-occurrences.

III. Selecting Discriminative Feature Co-occurrences using Sequential Forward Selection and Boosting

This section describes an algorithm for constructing a classifier for object detection by selecting discriminative feature co-occurrences. First, we define weak classifiers based on the co-occurrence of multiple features. Second, we describe a learning procedure based on stagewise selection of effective weak classifiers by boosting. Then we explain how discriminative feature combinations are found automatically. Two different versions of boosting algorithms are incorporated into the proposed framework: standard AdaBoost [13] and Real AdaBoost [8]. Other boosting algorithms such as LogitBoost [14] could be used instead. The standard AdaBoost algorithm is described as DAB (Discrete AdaBoost) to distinguish it from RAB (Real AdaBoost).

A. Weak Classifiers

This section defines weak classifiers based on feature co-occurrence.

First, we formulate these classifiers for Discrete AdaBoost. A function $J_t(x)$ represents an observation operation of feature co-occurrence from a sample image x in a weak classifier $h_t(x)$. When $J_t(x) = j$, based on the Bayesian decision rule, the classifier $h_t(x)$ is written as:

$$h_t(x) = \begin{cases} +1 & P_t(y=+1|j) > P_t(y=-1|j) \\ -1 & \text{otherwise} \end{cases}, \tag{3}$$

where $y \in \{+1, -1\}$ is the class label and $P_t(y=+1|j)$ and $P_t(y=-1|j)$ are class conditional probabilities of observing feature co-occurrence j from object (positive) and non-object (negative) classes respectively. They also represent joint probabilities of observing multiple feature combinations. They are evaluated with respect to the sample distribution D_t as follows:

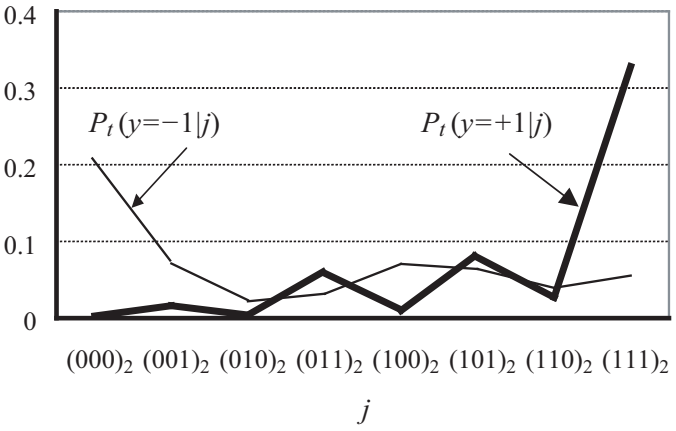

Fig. 6. A weak classifier based on class-conditional joint probabilities. $P_t(y=+1|j)$ and $P_t(y=-1|j)$ are obtained from three rectangle features. The three features yield eight combinations of binary variables. An input pattern is classified to be an object if $j=(011)_2$, $(101)_2$ or $(111)_2$.

$$P_t(y=+1|j) = \sum_{i:J_t(x_i)=j \wedge y_i=+1} D_t(i), \tag{4}$$

$$P_t(y=-1|j) = \sum_{i:J_t(x_i)=j \wedge y_i=-1} D_t(i), \tag{5}$$

where x_i is a sample image, $y_i \in \{+1,-1\}$ and $D_t(i)$ are the class label and the weight of the sample, respectively. Details are described in the next section.

An example of the probabilities $P_t(y=+1|j)$ and $P_t(y=-1|j)$ obtained from three rectangle features is shown in Figure 6. The three features yield eight combinations of binary variables, which are from $(000)_2$ to $(111)_2$. If $j=(011)_2=3$, $(101)_2=5$ or $(111)_2=7$ is measured from an input image, the input pattern is classified to be an object. If any other feature value is observed, it is classified as non-object.

Weak classifiers for Real AdaBoost are defined as follows:

$$h_t(x) = \frac{1}{2}\log\frac{P(y=+1|j)}{P(y=-1|j)}. \tag{6}$$

It may well happen that $P_t(y=-1|j)$ is very small or zero, in which case $h_t(x)$ will be infinite. To avoid this, we adopt the smoothing technique proposed in [8]:

$$h_t(x) = \frac{1}{2}\log\frac{P(y=+1|j)+\nu}{P(y=-1|j)+\nu}, \tag{7}$$

where ν is a small positive value. We set ν to $1/N$, where N is the number of training samples.

The weak classifiers for RAB return confidence scores which estimate the reliability of each of their predictions instead of binary values in (3) which indicate class labels as the classification results. The confidence scores evaluate differences between two probabilities. If the differences are large (which means j is discriminative), the scores become large and classification becomes more reliable. The weak classifier for DAB defined by (3) just compares the probabilities and determines which is larger. RAB improves the performance over DAB. We confirm this in the experiments carried out in Section IV-E.

1. Prepare a set of N labeled samples as $(x_i,y_i),\cdots,(x_N,y_N)$. $y_i \in \{+1,-1\}$ is the class label associated with the sample image x_i.
2. Initialize weights $D_1(i)=\frac{1}{N}$.
3. For $t=1,...,T$:
   (A) For each feature, calculate a feature value.
   (B) Binarize each feature value and assign a binary variable according to Eq.(1).
   (C) Train a weak classifier based on a combination of features.
   (D) Choose $h_t(x)$ with the lowest error ϵ_t. The error is evaluated with respect to the sample weight $D_t(i)$, $\epsilon_t = \sum_{i:y_i \neq h_t(x_i)} D_t(i)$.
   (E) Update the weights: $D_{t+1}(i) = \frac{D_t(i)\exp(-y_i\alpha_t h_t(x_i))}{\sum_i D_t(i)\exp(-y_i\alpha_t h_t(x_i))}$, where $\alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
   End For
4. Output the final strong classifier: $H(x) = sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$.

Fig. 7. Learning procedure based on DAB.

B. Learning Procedure based on Discrete AdaBoost (DAB)

The procedure for selecting features using DAB is shown in Figure 7. A set of N labeled training samples is given as $(x_1,y_1),\ldots,(x_N,y_N)$, where $y_i \in \{+1,-1\}$ is the class label associated with a sample x_i. $D_t(i)$ is a weight of a sample x_i. The weights are initialized by $D_1(i)=1/N$. The final strong classifier $H(x)$ is a linear combination of T weak classifiers $h_t(x)$:

$$H(x) = sign\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right), \tag{8}$$

where α_t are obtained from the error of each weak classifier weighted by D_t. At each stage of the boosting process, the best feature co-occurrence is selected according to steps (A) to (E) in Figure 7.

C. Learning Procedure based on Real AdaBoost (RAB)

Figure 8 shows the procedure for feature co-occurrence selection based on RAB. One difference from the procedure based on DAB is the criterion used for choosing weak classifiers in step (D). The criterion used in RAB is the Bhattacharyya bound Z_t on the sample distribution D_t, whereas that for DAB is an error rate ϵ_t. Another difference is the definition of the weak classifiers as in (3) and (7). The weak classifiers for RAB calculate the confidence scores and the reliability coefficients α_t for the weak classifiers are omitted from the final strong classifier.

D. Searching for Discriminative Feature Co-occurrence

To construct a weak classifier, we need to find discriminative feature co-occurrence. The best feature combination can be found

1. Prepare a set of N labeled samples as $(x_i, y_i), \cdots, (x_N, y_N)$. $y_i \in \{+1, -1\}$ is the class label associated with the sample image x_i.
2. Initialize weights $D_1(i) = \frac{1}{N}$.
3. For $t = 1, ..., T$:
   (A) For each feature, calculate a feature value.
   (B) Binarize each feature value and assign a binary variable according to Eq.(1).
   (C) Train a weak classifier based on a combination of features.
   (D) Choose $h_t(x)$ with the lowest Bhattacharyya bound Z_t: $Z_t = 2\sum_j \sqrt{P(y=+1|j)P(y=-1|j)}$. $P(y=+1|j)$ and $P(y=-1|j)$ are calculated by Eq.(4) and (5), respectively.
   (E) Update the weights: $D_{t+1}(i) = \frac{D_t(i)\exp(-y_i h_t(x_i))}{\sum_i D_t(i)\exp(-y_i h_t(x_i))}$.
   End For
4. Output the final strong classifier: $H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$.

Fig. 8. Learning procedure based on RAB.

by exhaustive search from all possible feature combinations. However, because the computational cost increases exponentially with the number of feature candidates, this is usually impractical. The computational complexity for selecting F from M feature candidates is $O(M^F)$. Branch-and-bound algorithms [28] can find the optimal solution more efficiently. However, when the number of feature candidates is very large, the worst case complexity is exponential. Furthermore, they require the feature selection criterion function to be monotonic, which cannot be satisfied in all cases. Several solutions for efficient feature selection have been proposed, but without a guarantee for optimal selection [27]. The best-known methods are Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS). SFS is a greedy approach starting with the best feature and adding other features one by one that satisfy a predetermined criterion. In contrast, SBS starts with all features as an initial subset and decreases it one by one. The Plus-l-Minus-r method [29] combines SFS and SBS, first adding l features to the subset using SFS and then deleting r using SBS. Sequential Forward Floating Selection (SFFS) and Sequential Backward Floating Selection (SBFS) [30] are generalizations of the Plus-l-Minus-r method, which automatically determine the values of l and r. Pudil at al. [30] compared these sequential methods and reported SFFS was performing best among them. In this paper, we use SFS because of its simple implementation. SFS adds features one by one to improve the classification accuracy. The accuracy is evaluated by ϵ_t for DAB or Z_t for RAB. The computational complexity becomes $O(FM)$. Pseudo-code for training weak classifiers based on SFS is shown in Figure 9.

How we determine the number of selected features F is also important. Choosing F too large leads to overfitting. Furthermore, the range of j doubles with each added feature. To avoid statistical

1. Initialize a feature subset: $S_0 = \emptyset$.
2. For $i = 1, ..., F$:
   For $j = 1, ..., NumOfFeatureCandidates$:
   (A) Generate a weak classifier by combining S_{i-1} with one feature f_j.
   (B) Evaluate accuracy of the weak classifier by a criterion function $G(S_{i-1} \cup f_j)$. G is evaluated by ϵ_t for DAB or Z_t for RAB.
   End For
   Select the best feature f^*: $f^* = \arg\min_j G(S_{i-1} \cup f_j)$.
   Add it to the subset: $S_i = S_{i-1} \cup f^*$.
   End For
3. Output the weak classifier with the subset S_F.

Fig. 9. Pseudo-code for training weak classifiers based on SFS. F features are selected through this procedure.

unreliability due to large histograms we limit F by,

$$2^{F^{max}} \times 10 < N, \tag{9}$$

so that at least 10 samples fall into each bin when each j is uniformly observed.

The following two methods for determining F are considered: (1) Select the best strong classifier from several classifiers trained using different values for F. Since a fixed F is used for each classifier, all weak classifiers observe the same number of features. The computational complexity for constructing each strong classifier is equivalent to that of Viola and Jones' framework. Therefore, the total cost for choosing the best classifier increases linearly according to the number of F settings.
(2) Choose the best F_t for each weak classifier h_t. The leave-one-out method or the bootstrap method can be used for the choice of F_t. However, in both cases training is expensive because repeated training and testing is required for estimating F_t. Here, we use the hold-out method, in which a set of samples is extracted from the training samples for validation. Each weak classifier tries to find the best number of combined features by incrementing F_t from 1 to F^{max}, so that the loss $L_{T'}$ calculated on the validation samples is minimized:

$$F_t = \arg\min_F L_{T'}. \tag{10}$$

The total cost for training the final strong classifier becomes F^{max} times larger than Viola and Jones' framework.

In order to reduce the time for training, F_t can be determined when $L_{T'}$ starts to increase before reaching F^{max}. In this case, $F_t + 1$ features need to be evaluated. The total cost for training becomes twice as high compared to Viola and Jones' framework in the worst case, where every weak classifier evaluates combinations of two features but chooses only one feature in the end.

$L_{T'}$ is defined as an exponential loss on the validation samples because the margin from the classification boundary can be considered by using the exponential loss instead of using an error rate. A set of N' labeled validation samples (x'_i, y'_i) is used for

calculating $L_{T'}$ by

$$L_{T'} = \frac{1}{N'} \sum_{i=1}^{N'} \exp\left(-y'_i H_{T'}(x'_i)\right), \tag{11}$$

where $H_{T'}(x)$ is a strong classifier at $t = T'$. It is represented as follows:

$$H_{T'}(x) = sign\left(\sum_{t=1}^{T'} \alpha_t h_t(x)\right), \tag{12}$$

where $h_{T'}$ is a weak classifier which is trained.

The first feature selected by the proposed framework combining SFS and DAB is the same as the one selected by Viola and Jones' framework. Other features may be different except for the worst case mentioned above, where all weak classifiers choose only one feature.

The hold-out method for choosing the number of features is applicable to training a cascade of classifiers using a slightly modified version of the algorithm in [7]. The training algorithm in [7] iterates adding one feature and evaluating the current classifier on the validation set in order to adjust the threshold. When the classifier satisfies the predetermined performance goal, e.g. a minimum detection rate and a maximum false positive rate, negative samples, which are classified to be positive by the current cascade, are collected for training the next classifier. Instead of adding one feature only by boosting, we use SFS or boosting. Features are added by SFS as long as the loss defined above on the validation set continues to decrease. Otherwise, features are added by boosting. The validation set, which is regarded as a hold-out set, can be commonly used for threshold adjustment and choosing the number of features.

E. Discussion

This section distinguishes our framework from previous approaches that utilize co-occurrence of multiple features.

The Local Binary Pattern (LBP) representation proposed by Hadid et al. [15] can express primitive features such as edges or corner points by comparing intensities between a target pixel and peripheral pixels. This approach uses co-occurrence of neighboring pixel pairs. However, only a limited number of feature combinations are evaluated. Our framework is able to use additional co-occurrences, which cannot be represented by these features by searching feature combinations at each stage of the boosting process.

Schneiderman and Kanade [6] learned the appearance of objects by evaluating dependencies between wavelet coefficients. Their face detector has one of the best reported detection performances [16]. However, it does not run in real-time since a large set of feature combinations given in advance is used for evaluating joint statistics. Our framework automatically selects a small number of discriminative feature combinations.

Template matching or PCA [17] based weak classifiers can also be incorporated into Viola and Jones' framework instead of using rectangle features, thus including spatial structure. Zhang et al. [18] select rectangle features at early stages of the boosting process and select the best eigenvector in later stages to improve accuracy. However, it is necessary to determine the switching stage based on the trade-off between speed and accuracy improvement because the computational cost of the PCA-based weak classifiers is larger than that of weak classifiers based on the rectangle features. In contrast, our framework does not increase the computational cost, whereas the accuracy is improved. Furthermore, their PCA-based weak classifiers use raw pixel data and are thus affected by changes in illumination which are not included in the training samples.

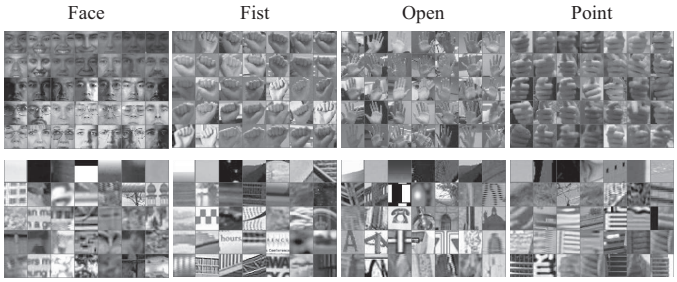

Fig. 10. Typical training samples. Top: positive samples, bottom: negative samples. The negative data includes difficult samples which are similar to the positive samples obtained during the bootstrapping process.

IV. Experimental Results

A. Data Collection

Figure 10 shows typical samples used in the experiments. We train four classifiers for detecting faces and hands in three different poses, respectively.

First, we explain how positive samples of the target objects were collected. Face samples were collected by extracting 10,000 images randomly from well-known public databases, which are AT&T [19], FERET [20], CMU-PIE [21], XM2VTS [23] and Yale [22]. Only frontal upright faces are selected from the databases. The face samples do not include pose variation but a variety of illumination conditions. The locations of pupils and nostrils are marked manually, and based on these points the face regions are aligned and scaled to a base resolution of 25×25 pixels. For hand data collection, we took video sequences in which people make three different hand gestures, which are called Fist (hand upright with curled fingers facing the camera), Open (open hand facing the camera) and Point (pointing toward the camera with the index finger) in this paper. The video sequences were taken in different illumination conditions. Top and bottom corners of hands are marked manually, and 5,000 hand samples are resized to 25×25 pixels. Each pose includes both left and right hands.

Next, we describe how negative samples were collected. We collected more than 8,000 images from the Web that do not contain any of the objects of interest. Each image was partitioned into patches of size 25×25 pixels and selected randomly so that the number of initial negative samples became 40% of the positive samples. There are 4,000 negative samples for the face classifier and 2,000 for the hand classifiers. Using this data, we trained Viola and Jones' classifiers. We added 2,000 non-face images and 1,000 non-hand images misclassified by these classifiers to the initial negative samples. We trained Viola and Jones' classifiers again using the updated data and collected misclassified samples until the number of negative and positive samples were equal. These bootstrapping [1] iterations were repeated three times in total. As a result many difficult negative samples similar to the positive samples were collected as shown in Figure 10.

TABLE I

FOUR CLASSIFIERS COMPARED IN THE FIRST EXPERIMENT. F IS THE NUMBER OF FEATURES USED IN EACH WEAK CLASSIFIER. T IS THE NUMBER OF WEAK CLASSIFIERS. THEREFORE, THE TOTAL NUMBER OF FEATURES FOR EACH STRONG CLASSIFIER IS CALCULATED BY $F \times T$. THE NUMBER OF WEAK CLASSIFIERS AND THE TOTAL NUMBER OF FEATURES ARE THE SAME ONLY IN $F1$, VIOLA AND JONES' CLASSIFIER. THE BOOSTING ITERATIONS ARE STOPPED WHEN THE TOTAL NUMBER OF SELECTED FEATURES REACHES $1{,}000$.

Classifiers	F	T	Total number of features
F1	1	1,000	1,000
F3	3	333	999
F5	5	200	1,000
F7	7	142	994

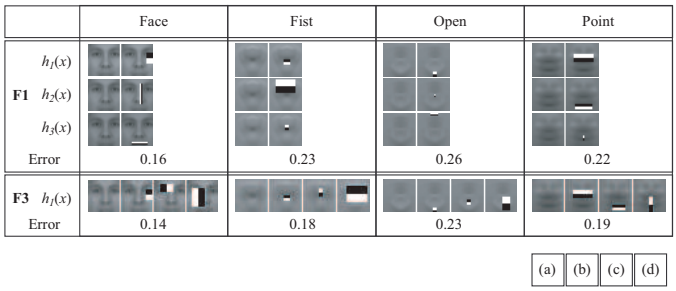

Fig. 11. Training results and error rates. The first three features selected for $F1$ (Viola and Jones') and $F3$ (ours) are shown. The first feature is the same but the subsequent features are different. $F3$ has a lower error rate than $F1$.

TABLE II

COMPUTING TIMES OF THE FOUR CLASSIFIERS. AVERAGE TIME FOR CLASSIFYING ONE SAMPLE WERE MEASURED ON AN INTEL XEON 3.2 GHZ PROCESSOR.

Classifiers	Total number of features	Seconds per sample
F1	1,000	0.000259
F3	999	0.000226
F5	1,000	0.000223
F7	994	0.000220

B. Performance Evaluation by Cross Validation

We adopt 10-fold cross validation for performance evaluation. The samples are partitioned into ten sets. Nine sets are used for training and the trained classifier is tested on the remaining set. The average error rate is calculated by repeating training and testing ten times on different combinations of sample sets. Error rates are calculated by counting the number of misclassifications.

C. Experiment (1): Performance Comparison between Classifiers Based on DAB

For each object four classifiers with different training parameters shown in Table I are trained using DAB and the basic feature set shown in Figure 1 (a). $F1$ indicates a classifier trained by Viola and Jones' framework and it is restricted to using a single feature for each weak classifier. The classifiers $F3$, $F5$ and $F7$ are trained by the proposed framework combining three, five and seven features, respectively, in each weak classifier. All weak classifiers use a predetermined number of features. An evaluation of choosing the number of features automatically is carried out in Experiment (2) described in the next section.

Figure 11 shows features selected for $F1$ and $F3$. Three features selected at the beginning of the boosting process are shown in (b), (c) and (d). Average images of object classes at sample distribution D_t are shown in (a). The k-th pixel value in the average image m_t is calculated by

$$m_{t,k} = \sum_i D_t(i) x_k(i), \tag{13}$$

where $x_k(i)$ is the k-th pixel value in the i-th sample image. The first feature is the same for $F1$ and $F3$, but the second and the third feature are different. This result is to be expected according to the algorithm shown above. The target objects have different structure and important features selected for classification are different from each other. For example, facial parts are important for detecting faces but silhouette information is more important for open hands. Error rates are also shown in the figure. The error rates of $F3$ are smaller than those of $F1$. This means that the co-occurrence of three features yields higher accuracy than the three features selected sequentially through the boosting process.

Figure 13 illustrates four error curves for each of the four objects. Error rates calculated by 10-fold cross validation are plotted against the number of features used for classification. The number of features is equivalent to the computational cost for classification when the basic feature set is used. All classifiers based on the proposed framework show smaller error rates than those of $F1$. Comparing the lowest error rates of $F1$ and $F3$, the error is reduced by about 30 to 50%. The error rates of $F5$ and $F7$ is higher than that of $F3$. This means that $F5$ and $F7$ overfit the training data because it combines many features in each weak classifier. Choosing the appropriate number of features F is thus important for improving the performance.

Table II shows the computing times of the four classifiers $F1$ to $F7$. Average classification times of one sample were measured on an Intel Xeon 3.2 GHz processor. The results are comparable. The methods using multiple features are slightly faster than $F1$. This is because the number of weak classifiers is smaller than that of $F1$ and thus the number of α_t terms is smaller.

To confirm the differences between the three frameworks explained in Figure 2, we carried out an additional experiment shown in Table III. The error rates of three classifiers, which consist of 15 features but combine them in different ways, are compared. $F1$ was trained according to Viola and Jones' framework, i.e., each weak classifier uses a single feature. $F15$ consists of only one weak classifier which combines 15 features by SFS. $F3$ was trained by our framework, which assigned three features for each of five weak classifiers. $F15$ yields the worst error rates because it overfits the training data and decreases statistical reliability due to combining too many features. $F3$ outperforms other classifiers.

In the subsequent experiments we compare $F1$ only with $F3$ because $F3$ has the lowest error rate among our three classifiers. Furthermore, we show that the proposed framework achieves higher performance when RAB is used for training instead of DAB and when the extended feature set is incorporated instead of the basic feature set. The final experiment evaluates robustness against occlusions.

TABLE III
ERROR RATE COMPARISON BETWEEN THREE CLASSIFIERS. EACH CLASSIFIER CONSISTS OF 15 FEATURES BUT WAS TRAINED IN DIFFERENT WAYS. WEAK CLASSIFIERS OF $F1$ DEPEND ON A SINGLE FEATURE. IT USES A SINGLE FEATURE FOR EACH OF 15 WEAK CLASSIFIERS. IN CONTRAST, $F15$ USES 15 FEATURES FOR ONLY ONE WEAK CLASSIFIER. $F3$ WAS TRAINED BY OUR FRAMEWORK, WHICH ASSIGNED THREE FEATURES FOR EACH OF FIVE WEAK CLASSIFIERS. $F15$ YIELDS THE WORST ERROR RATES BECAUSE OF OVERFITTING. ERROR RATES OF $F3$ ARE THE SMALLEST.

Classifiers	F	T	Face	Fist	Open	Point
F1	1	15	0.10	0.18	0.22	0.17
F15	15	1	0.12	0.19	0.24	0.21
F3	3	5	0.07	0.12	0.18	0.12

D. Experiment (2): Choosing the Number of Combined Features

In this experiment performances of three classifiers, $F1$, $F3$ and H are compared. $F1$ and $F3$ use a fixed number of features in each weak classifier. The classifier H automatically chooses the number of features using the hold-out method explained in Section III-D. The number of features is increased until the validation loss starts to increase or until reaching $F^{max} = 7$. Unlike other experiments, we cannot use 10-fold cross validation here because the number of features would be different for each weak classifier of the ten classifiers. We trained classifiers using four data sets out of ten and another two sets as the hold-out data for choosing the number of features. The classifiers were tested on the remaining four sets. For example, face classifiers were trained using 4,000 training samples and 2,000 hold-out samples and they were tested on the remaining 4,000 samples. For three hand classifiers, 2,000, 1,000 and 2,000 were assigned, respectively. All classifiers are based on DAB.

Figure 14 shows that the error rates of H and $F3$ are comparable and smaller than those of $F1$. We can conclude that choosing the number of features by the hold-out method is an effective way of improving the performance over $F1$.

E. Experiment (3): Performance Comparison between Classifiers Based on RAB

Figure 15 shows error rates of classifiers trained using RAB instead of DAB. Our classifier $F3$ consistently achieves smaller error rates than $F1$. This means that the proposed framework is still effective when a different version of the boosting algorithm is incorporated. Comparing Figure 13 with Figure 15, classifiers based on RAB perform better than those based on DAB.

F. Experiment (4): Performance Comparison between Classifiers Based on DAB with the Extended Feature Set

Figure 16 illustrates error rates of classifiers based on DAB using the extended feature set shown in Figure 1 (a) to (d) instead of the basic feature set, only (a). Note that the computational cost is not always equivalent to the number of features in this case, whereas they are the same when using the basic feature set. Our classifier $F3$ shows smaller error rates than those of $F1$. This means that the proposed framework is effective when the feature set is extended manually. Comparing Figure 13 with Figure 16, classification performance is greatly improved by using the extended feature set. Manual extension of the feature set is an effective way to construct a good classifier. Our framework further improves the performance.

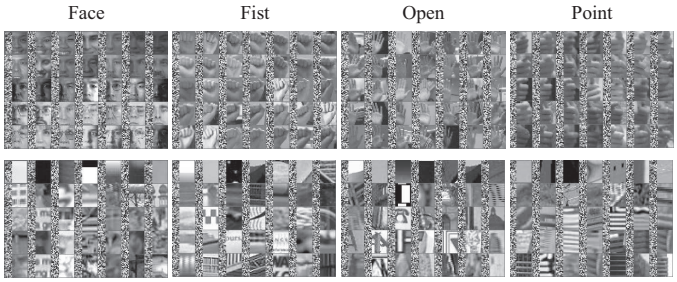

Fig. 12. Test samples partially occluded by random patterns. Top: positive samples, bottom: negative samples.

G. Experiment (5): Performance Comparison between Classifiers Based on RAB with the Extended Feature Set

Figure 17 illustrates error rates of classifiers based on RAB using the extended feature set. Our classifier $F3$ shows smaller error rates than those of $F1$.

H. Experiment (6): Evaluating Robustness against Occlusions

In this experiment robustness against occlusions is evaluated. The proposed method can be seen as using stronger spatial constraints for building weak classifiers compared to Viola and Jones' method. This brings up the question of whether we lose robustness towards occlusion. Again, the two classifiers $F1$ and $F3$ are compared. The classifiers are the same as those used in Experiment (5), which are trained with RAB on the extended feature set. The test samples are different from Experiment (5). As shown in Figure 12, they are partially occluded by random patterns. The size of the occluded areas is 8×25 pixels. About one third of the area of each sample is occluded. The intensity of each pixel in the occluding patterns is set by sampling a uniform distribution between 0 and 255. Figure 18 shows the error rates of $F1$ and $F3$. In this experiment the proposed method is not less robust towards occlusions. The performance against occlusions depends on the occluding patterns, the position and the size of the occluded areas and the target objects.

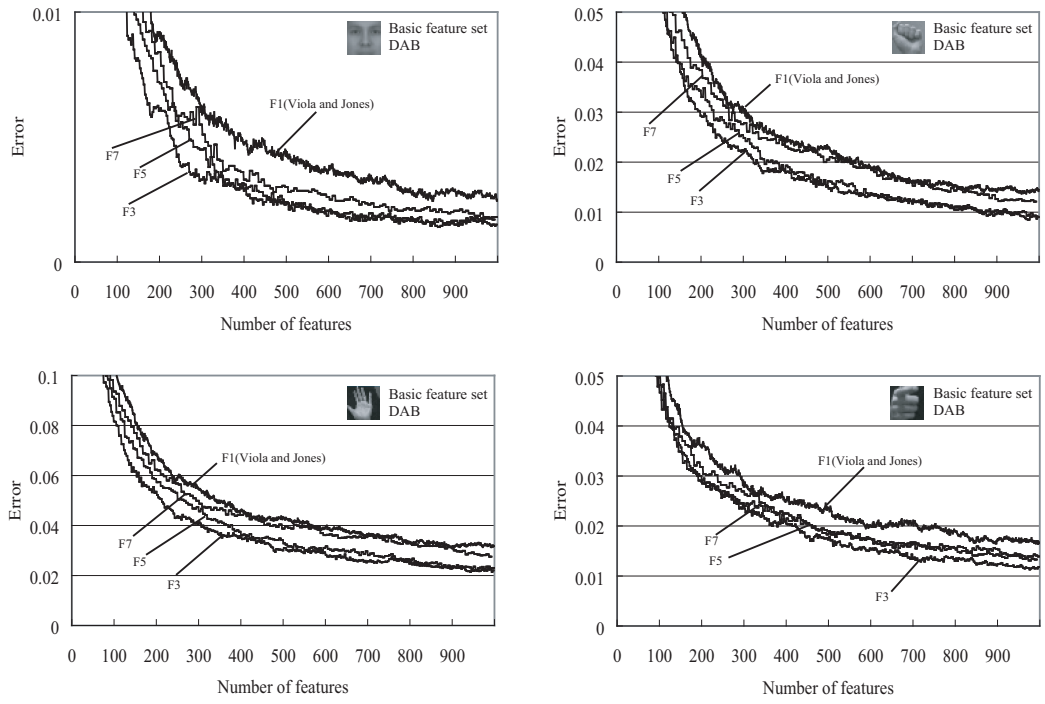

Fig. 13. Experiment (1): performance comparison of classifiers based on DAB. Top-left: performances of the Face detector, top-right: Fist, bottom-left: Open, bottom-right: Point.

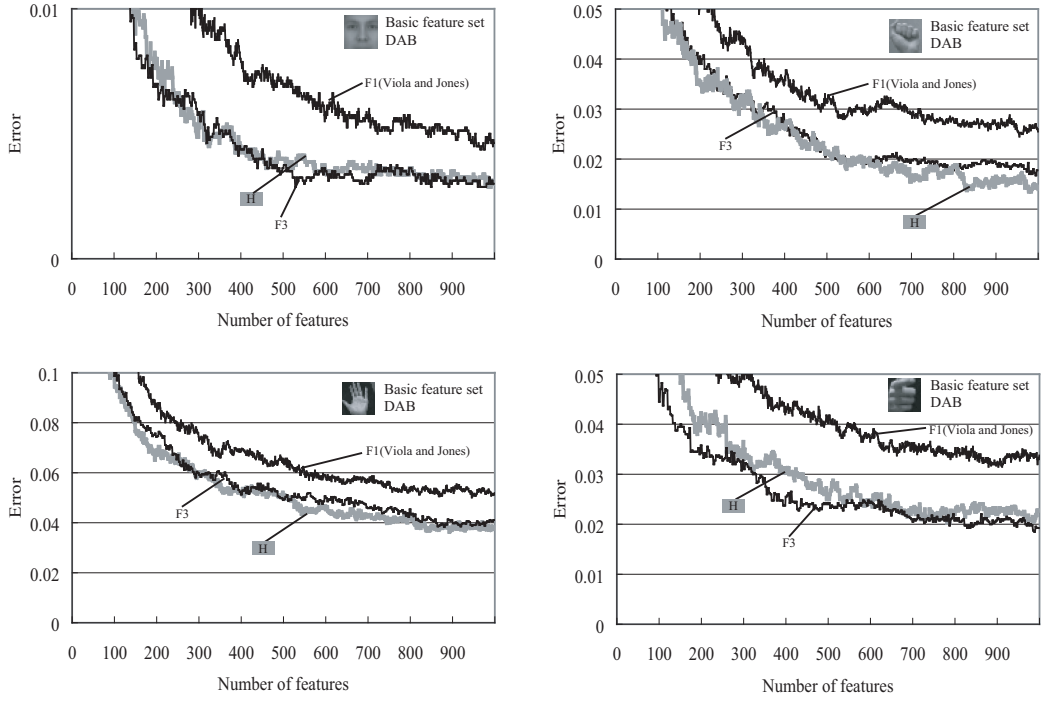

Fig. 14. Experiment (2): performance comparison of classifiers trained by the hold-out method. The number of features used in each weak classifier is automatically determined.

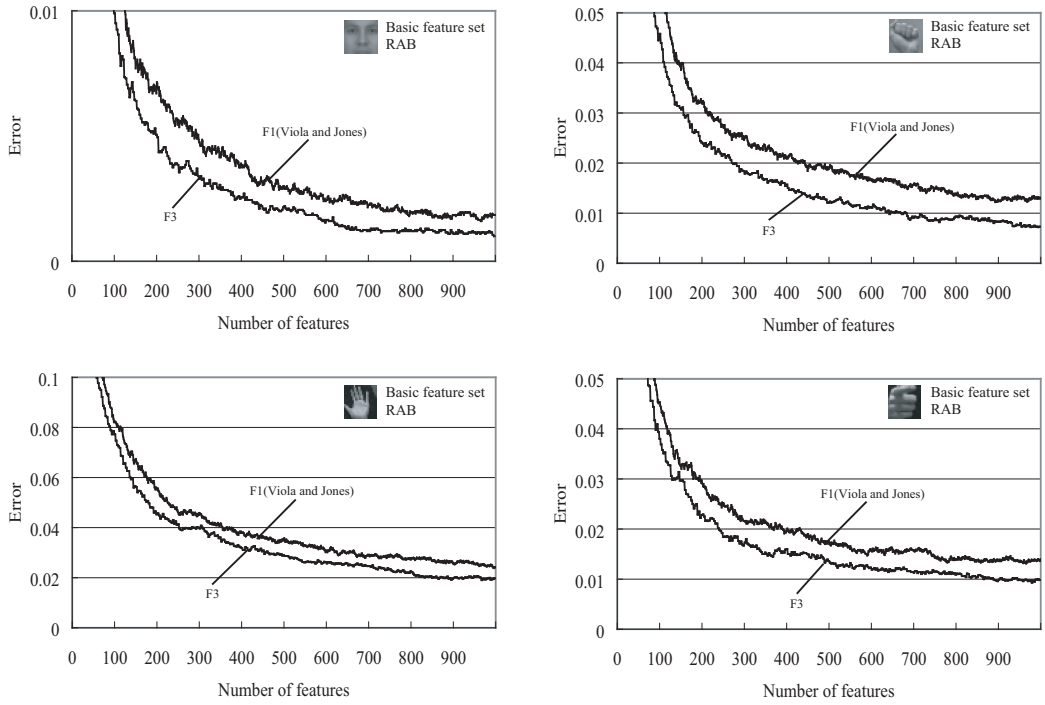

Fig. 15. Experiment (3): performance comparison of classifiers based on RAB.

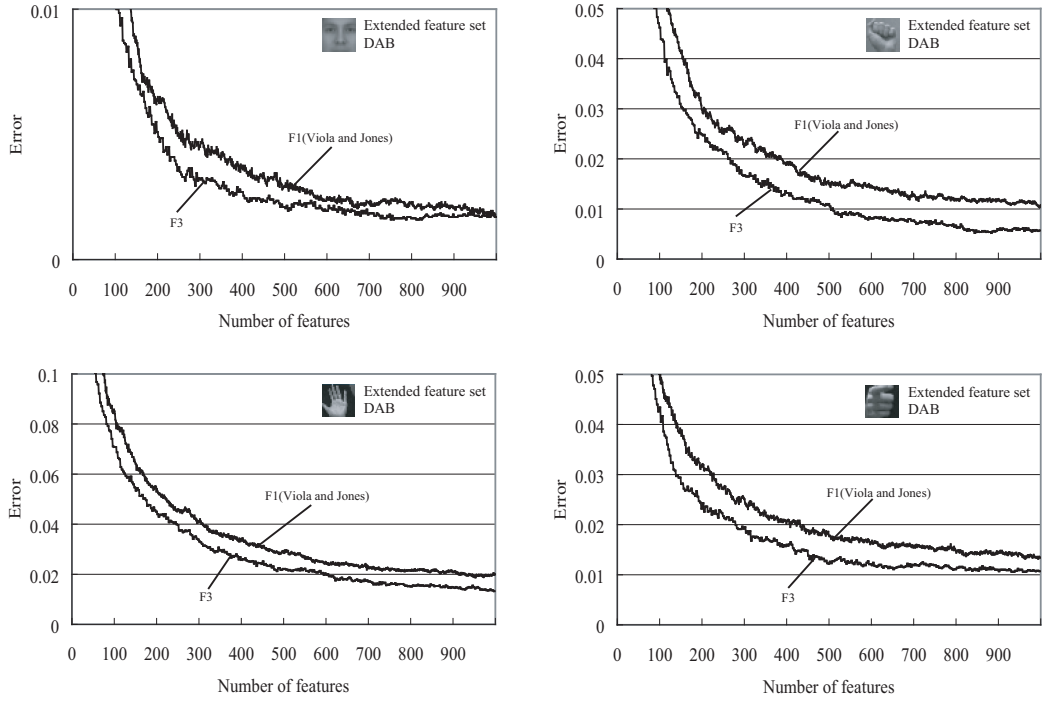

Fig. 16. Experiment (4): performance comparison of classifiers based on DAB with the extended feature set.

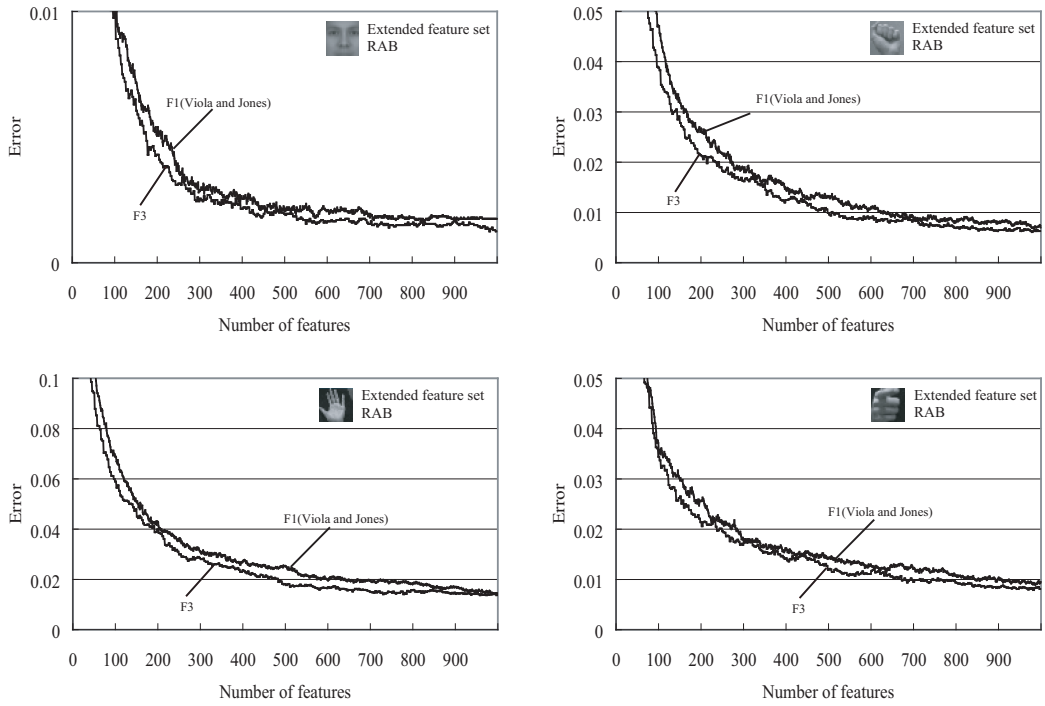

Fig. 17. Experiment (5): performance comparison of classifiers based on RAB with the extended feature set.

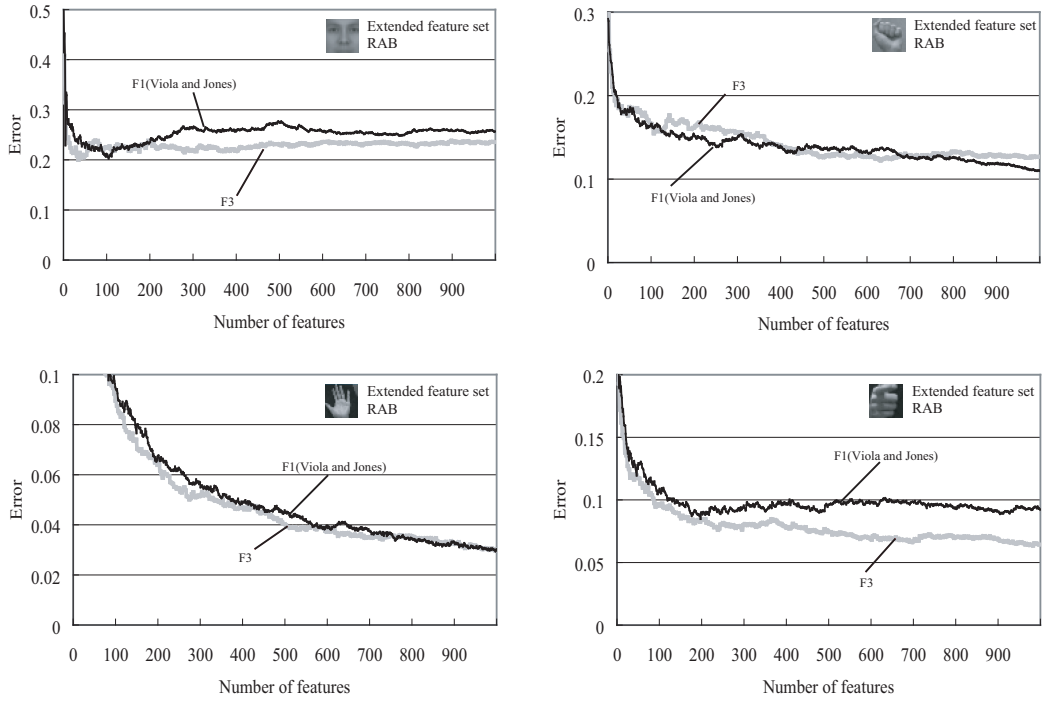

Fig. 18. Experiment (6): performance comparison in the presence of partial occlusion by random patterns. Classifiers based on RAB with the extended feature sets are used.

V. CONCLUSION

We have proposed a new framework for object detection. Experimental results show that the proposed framework yields higher performance than Viola and Jones' framework. The main advantages of the new framework are summarized as follows:

- It improves the classification power by incorporating co-occurrence of multiple features at the same computational cost. Alternatively, comparable accuracy is achieved at smaller computational cost.
- Co-occurrence of multiple features is expected to be useful for various kinds of objects because every object category exhibits some spatial structure.
- Different versions of boosting algorithms such as Real AdaBoost can be integrated with the proposed framework. Extensions of feature spaces can also be combined, possibly further improving the classification performance.

In this paper, we used only intensity gradient information extracted by rectangle features. In the future, we propose to investigate the integration of Gabor features or rectangle features in different feature spaces such as color or motion. The proposed framework could also be extended to multi-class object detection. Torralba et al. [31] proposed the idea of sharing features among multiple classes. Huang et al. [32] proposed the Vector Boosting algorithm which efficiently detects multiple object classes. We think that the proposed framework is applicable to these methods if there exist discriminative feature co-occurrences which can be shared among multiple classes.

REFERENCES

[1] K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.1, pp.39–51, 1998.

[2] H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.1, pp.23–38, 1998.

[3] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," Proc. IEEE Conf. CVPR, pp.130–136, 1997.

[4] B. Heisele, T. Poggio, and M. Pontil, "Face detection in still gray images," A.I. Memo no.1687, 2000.

[5] C. P. Papageorgiou and T. Poggio, "A trainable system for object detection," IJCV, vol.38, pp.15–33, 2000.

[6] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," IJCV, vol.56, pp.151–177, 2004.

[7] P. Viola and M. Jones, "Robust real-time face detection," IJCV, vol.57, pp.137–154, 2004.

[8] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," Mach. Learn., vol.37, pp.297–336, 1999.

[9] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on Real AdaBoost," Proc. IEEE Conf. Automatic Face and Gesture Recognition, pp.79-84, 2004.

[10] C. Liu and H. Y. Shum, "Kullback-Leibler boosting," Proc. IEEE Conf. CVPR, vol.1, pp.587–594, 2003.

[11] S. Z. Li and Z. Q. Zhang, "FloatBoost learning and statistical face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.9, pp.1112–1123, 2004.

[12] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," Proc. ICIP, vol.1, pp.900–903, 2002.

[13] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," In Computational Learning Theory: Eurocolt, pp.23–37, Springer-Verlag, 1995.

[14] J. Friedman, T. Hastie, and R. J. Tibshirani, "Additive logistic regression: a statistical view of boosting," Technical report, Department of Statistics, Sequoia Hall, Stanford University, July 1998.

[15] A. Hadid, M. Pietikäinen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," Proc. IEEE Conf. CVPR, vol.2, pp.797–804, 2004.

[16] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.1, pp.34–58 2002.

[17] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," J. Cognitive Neuroscience, vol.3, no.8, pp.71–86, 1991.

[18] D. Zhang, S. Z. Li, and D. G. Perez, "Real-time face detection using boosting in hierarchical feature spaces," Proc. ICPR, pp.411–414, 2004

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

[20] P. J. Phillips, H. Wechsler, J. Huang and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," Image and Vision Computing J, vol.16, no.5, pp.295–306, 1998.

[21] T. Sim, S. Baker and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database," Proc. of the 5th International Conference on Automatic Face and Gesture Recognition, 2002.

[22] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Generative models for recognition under variable pose and illumination," Proc. IEEE Conf. on Automatic Face and Gesture Recognition, pp.277–284, 2000.

[23] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," Proc. 2nd Conference on Audio and Video-based Biometric Personal Verification (AVBPA99), Springer Verlag, New York, 1999. URL: http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb

[24] E. J. Ong, and R. Bowden, "A boosted classifier tree for hand shape detection," Proc. IEEE Conf. on Automatic Face and Gesture Recognition, pp.889–894, 2004.

[25] M. Kölsch, and M. Turk, "Robust hand detection," Proc. IEEE Conf. on Automatic Face and Gesture Recognition, pp.614–619, 2004.

[26] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," IJCV, vol.63, pp.153–161, 2005

[27] A. Jain, and D. Zongker, "Feature selection: evaluation, application, and small sample performance," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.2, pp.153–158, 1997.

[28] P. M. Narendra, and K. Fukunaga, "A branch and bound algorithm for feature subset selection," IEEE Trans. Computers, vol.26, no.9, pp.917–922, 1977

[29] S. D. Strearns, "On selecting features for pattern classifiers," Proc. ICPR, pp.71–75, 1976.

[30] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," Pattern Recognit. Lett., vol.15, no.11, pp.1119–1125, 1994.

[31] A. Torralba, K. P. Murphy and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," Proc. of CVPR, 2004.

[32] C. Huang, H. Ai, Y. Li and S. Lao, "High-performance rotation invariant multiview face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.4, pp.671–686, 2007.