ORIGINAL ARTICLE

Histogram feature-based Fisher linear discriminant for face detection

Haijing Wang · Peihua Li · Tianwen Zhang

Abstract The face pattern is described by pairs of template-based histogram and Fisher projection orientation under the framework of AdaBoost learning in this paper. We assume that a set of templates are available first. To avoid making strong assumptions about distributional structure while still retaining good properties for estimation, the classical statistical model, histogram, is used to summarize the response of each template. By introducing a novel ''Integral Histogram Image'', we can compute histogram rapidly. Then, we turn to Fisher linear discriminant for each template to project histogram from d-dimensional subspace to one-dimensional subspace. Best features, used to describe face pattern, are selected by AdaBoost learning. The results of experiments demonstrate that the selected features are much more powerful to represent the face pattern than the simple rectangle features used by Viola and Jones and some variants.

Keywords Face detection · Histogram · Integral histogram image · Fisher linear discriminant · AdaBoost

H. Wang (✉) · T. Zhang
School of Computer Science and Technology,
Harbin Institute of Technology, P.O. Box 1071,
Harbin, Heilongjiang 150001, China
e-mail: ninhaijing@yahoo.com

P. Li
College of Computer Science and Technology,
Heilongjiang University, Harbin, Heilongjiang, China
e-mail: peihualj@hotmail.com

1 Introduction

Face detection is one of the visual tasks which humans can do effortlessly. Yet in computer vision community, this task is not easy. As a visual frontend processor, a face detection system should be able to achieve the task regardless of illumination changes, and orientation, position, scale, expression variations of human faces.

Viola and Jones [17] present the first highly accurate as well as real-time frontal face detector at 15 frames per second for 384 by 288 image. They use simple rectangle features to describe face pattern that can be computed rapidly by ''integral image''. Best features are selected automatically with AdaBoost learning, and cascade architecture is adopted to speed up detection. Many researchers present their works following the idea of Viola and Jones, mainly addressing two issues: (1) how to develop more powerful features to represent face pattern, and (2) how to classify samples based on the chosen representation.

From the view of feature selection, Murphy et al. [11] use a set of filters to convolve the image, and utilize the second and the fourth moments to calculate features from the different patches on filtered images. Levi and Weiss [7] take local edge orientation histograms (EOH) as features. Following the haar like features proposed by Viola and Jones [17, 18], other features include diagonal features [6], rotated features and center-surrounded features [9]. For the second issue, Wu et al. [21] propose a cascade learning algorithm based on forward feature selection which is two orders of magnitude faster than the Viola-Jones' approach and yields classifiers with similar quality. Li et al. [8] present the first real-time multiview face

detection system by FloatBoost. Torralba et al. [16] propose a multi-class boosting procedure (joint boosting) that reduces both the computation and sample complexity, by finding common features that can be shared across the classes. Using statistical detection theory, Zhou [25] constructs a binary decision tree to speed up the algorithm of Viola and Jones.

In this paper, we propose the novel feature, *template-based histogram along with Fisher projection orientation*, for face detection in the framework of AdaBoost learning. The results of experiments demonstrate that the selected features are much more powerful to represent the face pattern than the simple rectangle features used by Viola and Jones and some variants.

Our face detection algorithm consists of four major steps (see Fig. 1), as listed below.

1. *Image preprocessing:* Each training sample is scaled to 64 by 64 pixels, which includes enough rich information for template-based histogram calculation. Then, we take histogram equalization to make each image with equally distributed brightness levels over the whole brightness scale.

2. *Build template-based histogram feature set:* We assume that a set of templates are available first, then summarize the response of each template patch using one histogram, which represents marginal distribution of the patch. To speed up histogram computation, we extend "integral image" proposed by Viola and Jones [17] from one-dimensional to *d*-dimensional integral image, called "Integral Histogram Image" (IHI).

3. *Utilize Fisher linear discriminant to project histogram:* Fisher linear function yields the maximum ratio of between-class scatter to within-class scatter. Thus, for each template patch, we turn to Fisher linear discriminant to find a projection orientation of histograms. Two classes (faces and non-faces) are well separated by this Fisher projection orientation.

4. *Choose features by AdaBoost:* The best features to separate face and non-face samples are chosen by AdaBoost learning.

The paper is structured as follows. In Sect. 2, we present the template-based histogram feature set. Fisher linear discriminant is used to project histogram
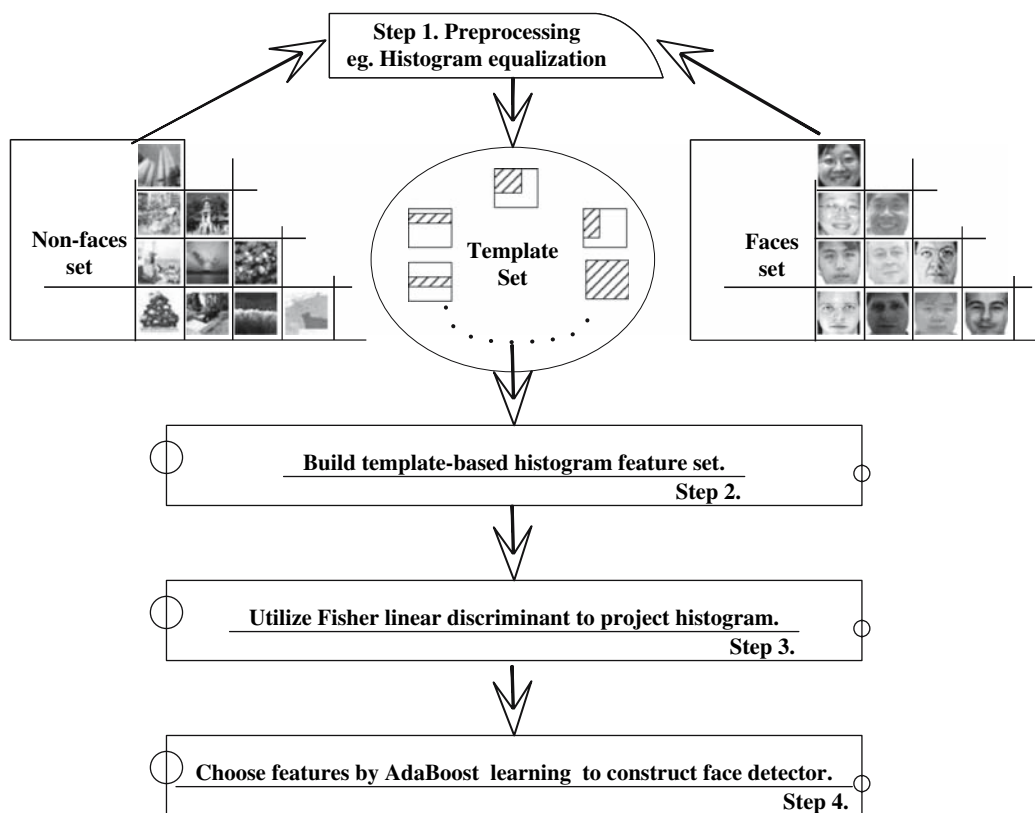

Fig. 1 Framework of face detection algorithm

in Sect. 3. The AdaBoost training to choose best features is described in Sect. 4. Experimental results are shown in Sect. 5. In Sect. 6, related works on face detection with histogram are discussed. Finally, conclusions and directions for future research are given.

2 Template-based histogram feature set

We assume that a set of reference patterns (templates) are available in this section. To seek statistical models of each template that avoid making strong assumptions about distributional structure while still retain good properties for estimation, the best compromise is histogram. To speed up histogram statistics, we extend "integral image" proposed by Viola and Jones [17, 18] from one dimension to d dimensions, called Integral Histogram Image (IHI).

2.1 Template set

Given a $p \times q$ image, any rectangle template t is a tetrad noted as $t = (x,y,w,h)$, where x and y are the location of horizontal and vertical coordinate of template t, respectively, and w and h represent width and height of template t, respectively. The rule is listed as follows.

i. Both width and height of each template are no less than eight pixels. And the step is eight pixels. It means $w \in \{8i|i = 1, 2, \ldots, \lfloor\frac{q}{8}\rfloor\}$ and $h \in \{8i|i = 1, 2, \ldots, \lfloor\frac{p}{8}\rfloor\}$.
ii. The rectangle templates are created in a step of eight pixels along both horizontal and orthogonal direction. There are $x \in \{8i|i = 0, 1, \ldots, \lfloor\frac{q}{8}\rfloor - 1\}$ and $y \in \{8i|i = 0, 1, \ldots, \lfloor\frac{p}{8}\rfloor - 1\}$.
iii. Each template should satisfy that $x + w \leq q$ and $y + h \leq p$.
iv. Each template includes $p \times q/16$ pixels at least.

According to the rule above, there are 1,024 templates for 64×64 image. In repeated experiments, we find that 132 templates are less chosen during the AdaBoost training. Therefore, we choose 892 different rectangle templates from the total 1,024 ones to form the final template set to speed up training. Taking the

top left point $(x,y) = (0,0)$, for example, Fig. 2 shows the corresponding 59 reference patterns by rule (i)–(iv). The black rectangles are the masks used to calculate histogram features.

2.2 Integral image

The integral image was first proposed by Viola and Jones [17, 18], the advantage of which is that the sum of all pixel intensity of any rectangle region in an image can be calculated at negligible cost. The integral image at location (x,y) contains the sum of the pixel intensity above and to the left of (x, y), inclusive:

$$\xi(x, y) = \sum_{x' \leqslant x, y' \leqslant y} I(x', y') \tag{1}$$

where $\xi(x,y)$ is the integral image and $I(x,y)$ is the gray value of the original image at location (x,y). The algorithm of integral image is presented as follows.

Input: Original image $I(x,y)$ with $p \times q$ pixels.
Algorithm:

i. Initialize the auxiliary line corresponding to the first row and the first column: $\xi(x,0) = 0$, $x = 0,\ldots, q$ and $\xi(0,y) = 0$, $y = 1,\ldots, p$.
ii. Repeat for $y = 1,\ldots, p$: Repeat for $x = 1,\ldots, q$: $\xi(x,y) = \xi(x,y{-}1) + \sum_{k=1}^{x} I(k,y)$.

Output: Integral image $\xi(x,y)$ corresponding to $(p + 1) \times (q + 1)$ matrix.

The sum of all pixel intensity of an upright rectangle $r(x,y,w,h)$ (see Fig. 3a, b) can be determined by (2).

$$\xi(r) = \xi(x+w, y+h) - \xi(x+w, y) - \xi(x, y+h) + \xi(x, y) \tag{2}$$

Next, we will extend the concept of integral image to "integral histogram image".

2.3 Integral Histogram Image

Motivated by integral image of Viola and Jones, we present "Integral Histogram Image" (IHI) [13, 19] (see Fig. 3) through which histogram of any rectangle region in an image can be computed via array index operations. Given a $p \times q$ gray image, the IHI Υ is a
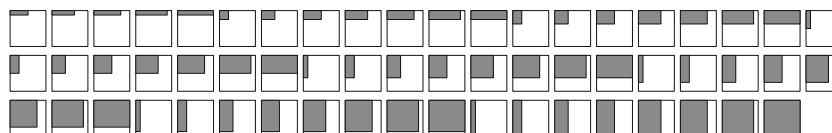

Fig. 2 Example templates with the top left point $(x,y) = (0,0)$ for $p \times q = 64 \times 64$ image. The *black rectangles* are the masks used to calculate histogram features

Fig. 3 Integral image versus integral histogram image. Based on the same rectangle region shown in (**a**), (**b**) gives the integral image proposed by Viola and Jones [17]; and our integral histogram image is shown in (**c**)

(a) Coordinate of Rectangle

(b) Integral Image (Viola and Jones)

(c) Integral Histogram Image (Our Approach)
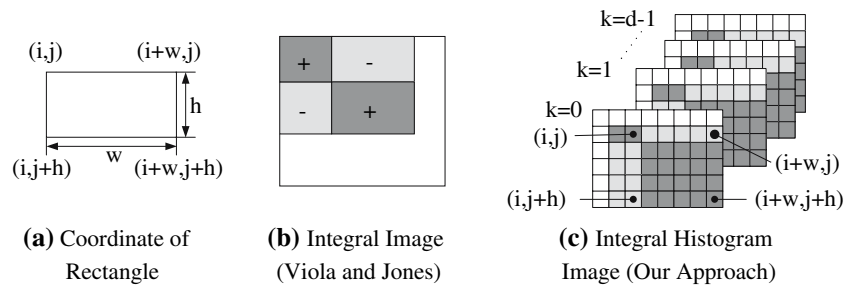
three-dimensional array as $\Upsilon[p+1][q+1][d]$, where d is the number of histogram bins. The algorithm of integral histogram image is described as two steps.

i.   For a $p \times q$ gray image, create a three-dimensional array $\Upsilon[p+1][q+1][d]$ initialized with 0, where d is the number of histogram bins.
ii. Form the cumulative histogram image. Repeat for $i = 1, ..., p$: Repeat for $j = 1, ..., q$:

$$\alpha[k] \leftarrow \alpha[k] + \delta(i,j), \quad k = 0, 1, \ldots, d-1$$

where $\delta(i,j) = 1$ if the intensity of pixel at location (i,j) belongs to the kth bin of histogram; otherwise $\delta(i,j) = 0$. If $j = 1$, $\alpha[k]$ is first set with 0 for $k = 0, 1, ..., d–1$ before continuing the above operation. Next, the IHI increases the relevant member of Υ.

$$\Upsilon[i][j][k] = \Upsilon[i-1][j][k] + \alpha[k]$$

where $k = 0, 1, ..., d–1$. ∎

The IHI can be computed in one pass over the original image. At location (i,j), the IHI $\Upsilon[i][j][k]$ corresponds to the number of pixels falling into the kth bin, the spatial location of which is above and to the left of (i,j). The histogram $h_r[k]$ ($k = 0, ..., d–1$) of any rectangle region $r(i,j,w,h)$ can be determined in $(4 \times d)$ array references by IHI (see Fig. 3c):

$$h_r[k] = \Upsilon[i+w][j+h][k] - \Upsilon[i+w][j][k] \\ - \Upsilon[i][j+h][k] + \Upsilon[i][j][k] \qquad (3)$$

where $k = 0, ..., d–1$, w and h are the width and height of rectangle r, respectively.

3 Histogram projection by Fisher linear discriminant

Different from principal component analysis (PCA), which seeks directions efficient for representation, Fisher linear discriminant seeks directions efficient for discrimination by yielding the maximum ratio of between-class scatter to within-class scatter. Thus, we

turn to Fisher linear discriminant for each template to find a projection orientation of histograms by which two classes (positives and negatives) are well separated. Each template is corresponding to one Fisher projection orientation, while the classification task can be converted from a d-dimensional (each histogram includes d bins) to a one-dimensional problem.

For any one template defined in the previous section, there is a set $\mathcal{H}$ including n histograms, each of which is divided into d bins.

$$\mathcal{H} = \mathcal{H}_1 \bigcup \mathcal{H}_2 = \{\mathbf{h}_1, \ldots, \mathbf{h}_n\}, \quad n_1 + n_2 = n \qquad (4)$$

where n_1 and n_2 are cardinality of subset $\mathcal{H}_1$ labeled τ_1 (positives) and subset $\mathcal{H}_2$ labeled τ_2 (negatives), respectively. If we form a linear combination of the components of $\mathbf{h}_i$, we obtain the scalar dot product

$$z_i = \mathbf{v}^t \mathbf{h}_i \qquad (5)$$

and a corresponding set $\mathcal{Z} = \{z_1, \ldots, z_n\}$ of n projected points divided into the subsets $\mathcal{Z}_1$ and $\mathcal{Z}_2$. Geometrically, if $\|\mathbf{v}\| = 1$, each z_i is the projection of the corresponding $\mathbf{h}_i$ onto a line in the direction of $\mathbf{v}$.

The Fisher linear discriminant employs the linear function [see (5)] for which the criterion function

$$J(\mathbf{v}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \qquad (6)$$

is maximized. That is, the $\mathbf{v}$ maximizing $J(\cdot)$ leads to the best separation between the two projected sets ($\mathcal{H}_1$ and $\mathcal{H}_2$). Here $\tilde{m}_i$ ($i = 1, 2$) is the mean of the projected points set $\mathcal{Z}_i$ corresponding to the histograms set $\mathcal{H}_i$. We define the scatter for projected histograms labeled τ_i by

$$\tilde{s}_i^2 = \sum_{z \in \mathcal{Z}_i} (z - \tilde{m}_i)^2, \quad i = 1, 2 \qquad (7)$$

Thus, $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ is an estimate of the variance of all histograms, and $\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total within-class scatter of the projected histograms.

According to the generalized Rayleigh quotient well known in mathematical physics, the criterion function $J(\cdot)$ in (6) can be written as

$$J(\mathbf{v}) = \frac{\mathbf{v}^t \mathbf{S}_B \mathbf{v}}{\mathbf{v}^t \mathbf{S}_W \mathbf{v}} \qquad (8)$$

$\mathbf{S}_W$ is called the within-class scatter matrix defined by

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2, \quad \mathbf{S}_i = \sum_{\mathbf{h} \in \mathcal{H}_i} (\mathbf{h} - \mathbf{m}_i)(\mathbf{h} - \mathbf{m}_i)^t, \quad i = 1, 2 \qquad (9)$$

where $\mathbf{m}_i$ is the mean of all elements of set $\mathcal{H}_i$. $\mathbf{S}_B$ is called the between-class scatter matrix defined by

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t \qquad (10)$$

Now we get the solution for the projection orientation $\mathbf{v}$ that optimizes $J(\cdot)$ as:

$$\mathbf{v} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \qquad (11)$$

which is sometimes called the canonical variance.

Thus the classification problem of d-dimensional subspace is converted to a hopefully more manageable one-dimensional one by projecting template-based histograms onto a line by (5) and (11) for subsequent AdaBoost learning.

4 Feature selection by gentle AdaBoost algorithm

Boosting algorithm, proposed in the Computational Learning Theory literature [2], is a method to find a highly accurate hypothesis (a strong classifier) by combining many "weak" hypotheses, each of which is based on the reweighted version of the training data in order to emphasize those which are incorrectly classified by previous weak classifiers, and only moderately accurate. The final strong classifier is a weighted combination of weak classifiers followed by a threshold. The adaptive version of Boosting is called Ada-Boost [3]. We and others [9] have found that Gentle AdaBoost (GAB) [4] gives higher performance than Discrete AdaBoost (DAB) [3] and Real AdaBoost (RAB) [15], and requires fewer iterations to train. Instead of DAB used by Viola and Jones [17, 18], GAB is used both to select features and to train the classifier. We will briefly present GAB below.

Given a training set $\mathcal{X}$ with its weight distribution D, the Boosting procedure computes a weak hypothesis $f : \mathcal{X} \mapsto R$, where the sign of f is the predicted label $\lambda \in \{\tau_1, \tau_2\}$ of the sample $x \in \mathcal{X}$, and the magnitude $|f(x)|$ is the confidence in this prediction. This is called RAB [15]. The simplest case, $f : \mathcal{X} \mapsto \{-1, +1\}$, is called

DAB [3]. Let $f_1, f_2, ..., f_T$ stand for a set of learned weak hypotheses, thus the ensemble hypothesis is

$$F(x) = \mathcal{E}[\lambda | x] = \sum_{t=1}^{T} f_t(x) \qquad (12)$$

where $\mathcal{E}$ represents the expectation.

Suppose, we have a current estimation F and seek an improved estimation $F + f$ by the minimizing criterion $J(F + f) = \mathcal{E}[e^{-\lambda(F(x)+f(x))}]$. RAB optimizes J with respect to $f(x)$ at each iteration. GAB [4], a modified version of RAB, takes adaptive Newton steps to minimize $J(F + f)$ by (13).

$$F(x) \leftarrow F(x) + \frac{\mathcal{E}[e^{-\lambda F(x)} \lambda]}{\mathcal{E}[e^{-\lambda F(x)}]} = F(x) + \mathcal{E}_\omega[\lambda | x] \qquad (13)$$

Here, the notation $\mathcal{E}_\omega[\lambda | x]$ refers to a weighted conditional expectation, and the weight is updated by (14).

$$\omega \leftarrow \omega \cdot e^{-\lambda f(x)} \qquad (14)$$

Therefore, the weak hypothesis $f(x)$ is written as

$$f(x) = \mathcal{E}_\omega[\lambda | x] = \frac{\mathcal{E}[e^{-\lambda F(x)} \lambda]}{\mathcal{E}[e^{-\lambda F(x)}]} \qquad (15)$$

To get optimized $f(x)$, we expand $J(F + f)$ to the second order about $f(x) = 0$. Minimizing pointwise with respect to $f(x)$, there is

$$\hat{f}(x) = \arg \min_f \mathcal{E}_\omega[(\lambda - f(x))^2 | x] \qquad (16)$$

Equation (16) shows the way to obtain the weak hypothesis $f(x)$.

We restrict the weak classifier $f(x)$ shown in (16) to the set of classification functions, each of which depends on a single feature, template-based histogram Fisher projection orientation, described in the previous section. The weight distribution of samples is updated via (14) at each round of GAB learning. The category of any given sample x is decided by the sign of the strong classifier $F(x)$ [see(12)], as shown in (17).

$$C(x) = \text{sign}[F(x)] = \text{sign}\left[\sum_{t=1}^{T} f_t(x)\right] = \begin{cases} +1 & \text{if } F(x) \geq \vartheta \\ -1 & \text{otherwise} \end{cases} \qquad (17)$$

The output of $C(x)$ with +1 or –1 represents the sample x belongs to positives or negatives, respectively. And threshold ϑ is decided by the prescribed hit rate of the strong classifier $F(x)$ to positives. The hit rate of a classifier is estimated as (18) in [1].

$$\varrho \approx \frac{\text{Positives correctly classified}}{\text{Total positives}} \qquad (18)$$

Suppose there is n_1 positives, the set of positives $\Delta = \{x_1, x_2, \ldots, x_{n_1}\}$ is a subset of the training set $\mathcal{X}$. For the set Δ, there is a group of output of the strong classifier $F(x) = \sum_{t=1}^{T} f_t(x) : F(x_1), F(x_2), \ldots, F(x_{n_1})$. They are sorted in descending order shown in (19)

$$F(x_{k(1)}) \leqslant F(x_{k(2)}) \leqslant \cdots \leqslant F(x_{k(n_1)}) \qquad (19)$$

where $i \rightarrow k(i)$ is a mapping by descending order $F(x)$. Among all output of positives, we will choose one of them as the threshold ϑ of the strong classifier $F(x)$ by the hit rate ϱ, as shown as Eq. (20).

$$\vartheta = F(x_{k(\lfloor (1-\varrho) \times n_1 \rfloor)}) \qquad (20)$$

For example, the hit rate $\varrho = 0.99$ means $\lceil \varrho \times n_1 \rceil = \lceil 0.99 n_1 \rceil$ positives are classified correctly according to $C(x)$ [see(17)], and $\lfloor (1-\varrho) \times n_1 \rfloor = \lfloor 0.01 n_1 \rfloor$ positives are classified as negatives.

A cascade of strong classifiers is also constructed to increase the speed of the detector by focusing attention on promising regions of the image. Simpler classifiers are used to reject the majority of sub-windows before more complex ones are called upon to achieve low false positive rates [17, 18].

5 Experimental results

In this section, we first introduce the training data set and feature set. Then learning results and detection results are described.
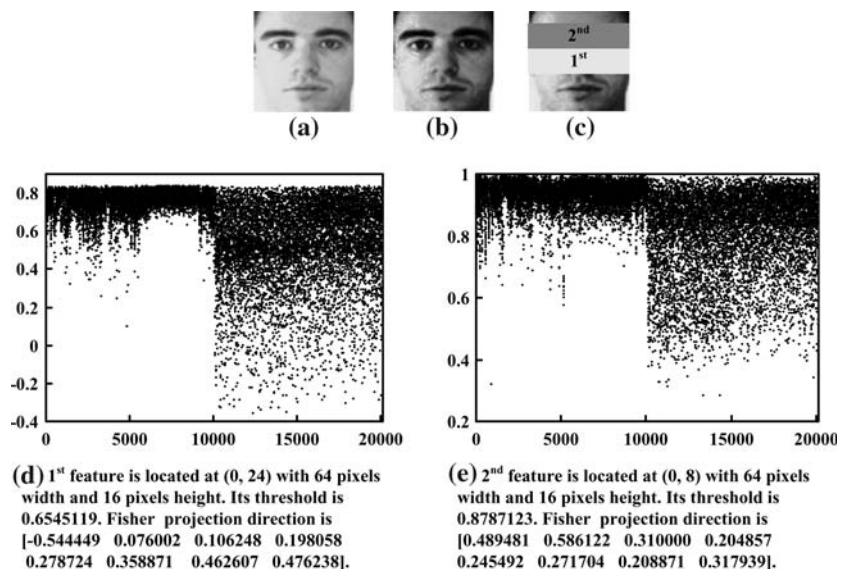
We crop 10,135 frontal face images as training positives. The negative samples are collected by selecting random sub-windows from a set of 24,621 images which do not contain faces. For each layer training, the maximum size of the negatives set is 10,000. Each sample is scaled to 64 by 64 pixels, which includes enough rich information for template-based histogram calculation. We take histogram equalization for both training samples and test samples to make each image with equally distributed brightness levels over the whole brightness scale. Figure 4b is an example of histogram equation to the original image shown in Fig. 4a.

Given the base resolution of the detector is 64×64, our feature set only includes 892 template-based histograms features (refer to Sect. 2.1 for more details on the template set), which is far less than 45,396, the size of feature set of Viola and Jones' 24×24 detector. We calculate histogram with eight bins at each template location. So the corresponding Fisher projection orientation is also an eight dimensional vector.

The first and second features of our detector are shown in Fig. 4d and e shows projection values of all training samples on the first two Fisher linear discriminant orientations, [–0.544449 0.076002 0.106248 0.198058 0.278724 0.358871 0.462607 0.476238] and [0.489481 0.586122 0.310000 0.204857 0.245492 0.271704 0.208871 0.317939]. At threshold 0.6545119 and 0.8787123, 99% positives are classified correctly after the first and the first two features are chosen by Gentle AdaBoost learning. At each round training, positives is trimmed according their weights to emphasize difficult samples.

Our cascade detector only includes 17 layers with 2,347 features. Note that the final detector of Viola and

Fig. 4 (**a**) and (**b**) are the original positive image and the image after histogram equation, respectively. (**c**) Shows the locations of the first and second features. The projections of all training samples corresponding to the first and second features of the detector are shown in (**d**) and (**e**). X axis represents the sample ID. The first 10,135 samples are positives (faces) and the Id from 10,136 to 20,135 represents negatives (non-faces). Y axis is the Fisher linear projection value

(**d**) 1st feature is located at (0, 24) with 64 pixels width and 16 pixels height. Its threshold is 0.6545119. Fisher projection direction is [-0.544449 0.076002 0.106248 0.198058 0.278724 0.358871 0.462607 0.476238].

(**e**) 2nd feature is located at (0, 8) with 64 pixels width and 16 pixels height. Its threshold is 0.8787123. Fisher projection direction is [0.489481 0.586122 0.310000 0.204857 0.245492 0.271704 0.208871 0.317939].

Jones [18] is a 38 layer cascade of classifiers which includes a total of 6,060 features.

For Viola and Jones' approach, training time for the entire 38 layer detector is on the order of weeks on a single 466 MHz AlphaStation XP900. Utilizing novel ''IHI'' and our small feature set (892) compared with the size of Viola and Jones (45,396), our training process can be finished in 2 days on a single Pentium 4 CPU 3.00 GHz. IHI saves one-third times for both training and detection.

To further demonstrate the powerfulness of the proposed novel features, we train a cascaded classifier containing ten 20-feature classifiers as done in Viola and Jones' experiment [18]. The detector is trained using 5,000 faces and 10,000 non-face sub-windows randomly chosen from non-face images. Then, we give the ROC curves to compare the performance of our detector with the detector of Viola and Jones in Fig. 5. It is shown that the template-based histogram representation is a good choice for face detection and yields results that are better than Viola-Jones' method.

The detector scans the image at multiple scales and locations. And the test set is the CMU face test set [14] without containing images with line drawn faces. The detection rate achieves 90% with 86 false detections. ROC curve is shown in Fig. 6. Some typical detection results are given in Fig. 7.

More time consuming is the preprocessing of image to help correcting the lighting disparities, which takes about half of the total processing time. So our future work will focus on this direction to speed up our approach. As the current detection computations are intrinsically parallel, another possible improvement is to implement our approach on parallel machines or
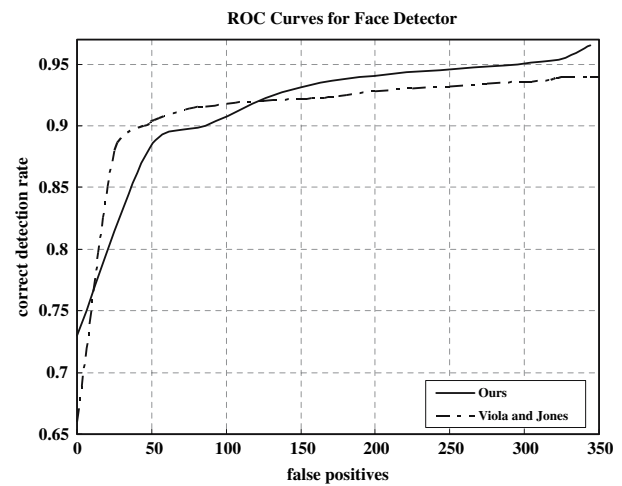

Fig. 6 ROC curves for proposed face detector and Viola and Jones' detector on the CMU test set

special-purpose chips. The speed of detection also depends on the number of histogram bins. When the number is large, the speed might be slow. The current histogram range is divided equally into d units. One way to improve the expression of the features is to choose the histogram boundary adaptively.

6 Related work and discussion

Our work takes template-based histogram and corresponding Fisher projection orientation as feature for face detection. Under the framework of face detection with histogram, we classify related works into four groups: spectral histogram, orientation histogram, histogram divergence, and LBP-based histogram. Since color histogram is commonly taken for object detection in color image, our discussion does not include it.

6.1 Spectral histogram

Waring and Liu [20] present a face detection method using spectral histograms of filter images as features and support vector machines (SVMs) as classifiers. The spectral histogram of an image is defined as the concatenation of histograms for every filtered image. Compared with their work, our feature is template-based histogram which holds the space location information. In essence, their feature set is based on one template of the whole image which is far less than our defined 892 templates. Their method achieves the best performance with respect to both false detections (67) and detection rate (96.7%) on CMU test set. However, it is at the cost of detection time. It takes several minutes to process a typical test image.
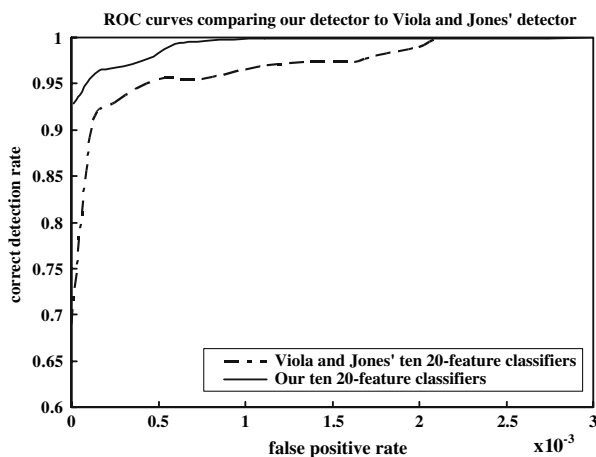

Fig. 5 ROC curves comparing a cascaded classifier containing ten 20-feature classifiers by Viola and Jones with proposed cascaded classifier

Fig. 7 Output of our face detector on a number of test images from the CMU new test set

6.2 Orientation histogram

Zhou et al. [24] use the orientation of the principal axis and a local peak at its orthogonal orientation to investigate the orientation histogram. They separate an upright human face into 3×3 blocks, the orientation histogram of which will satisfy three criteria. The resolution of orientation can be adjusted by bin of histogram. However, it is a coarse detection without combining with other face detection methods for the absence of enough features.

Levi and Weiss [7] present local edge orientation histograms (EOH) as features and AdaBoost used to construct face classifier. The histogram is transferred from d-dimensional vector into a scalar value by the ratio value of any two bins within some regions. We take Fisher discriminant to project histogram features before AdaBoost learning. It makes the divergence of

feature values better than the case without Fisher projection.

6.3 Histogram divergence

Liu and Shum [10] propose Kullback–Leibler (KL) Boosting and use the histogram divergences of two classes on these linear features as evidence for classification. To choose features for classifier, they maximize the KL divergence of histograms of positive and negative samples projected on the features. Note that KL divergence corresponds to our Fisher discriminant.

6.4 LBP-based histogram

Local binary pattern (LBP) operator [12] is a measure of the spatial structure of local image texture. The face image is represented by a concatenation of a global and

a set of local LBP histograms in [5]. The local and global LBP histograms are computed by dividing the face image into several overlapping blocks and over the whole face image, respectively. The limit of concatenation of histograms is that each sub-histogram must contribute to the decision of classifier. Different with their work, we utilize AdaBoost to choose important templates to construct classifier.

Based on LBP images, Zhang and Zhao [22] extract five measurements from original color images, and compute histograms in 23 spatial templates. The coarse detection takes histogram intersection as similarity measurement between the average face histogram of training samples and histogram of the test sample. Next all $23 \times 5 = 115$ histograms are concatenated to construct the input of SVM classifier. In their consecutive work [23], spatial histogram based on template is introduced to represent a LBP image. Fisher criterion measures discriminating ability of the distance (histogram intersection) of each pair of histograms. It means the input of Fisher is a scalar value. In our method, the vector of histogram is the input of Fisher discriminant. Which discriminant ability is better to histogram projection with or without histogram distance calculation? It deserves further discussion. However, it is out of the scope of this paper.

7 Conclusions

Fisher linear discriminant is used to project template-based histograms features for the task of face detection in this paper. We choose best features, pairs of template-based histogram along with Fisher projection orientation, by AdaBoost algorithm. The experimental results demonstrate that the selected features are very powerful to describe the face pattern. There are a number of directions for future work, including adaptive selection of histogram dimensions, extending the framework to multi-view face detection, and employing more sophisticated image preprocessing and normalization techniques.

References

1. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27:861–874
2. Freund Y, Schapire RE (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. pp 23–37
3. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. Machine learning. In: Proceedings of the thirteenth international conference. pp 148–156
4. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 28:337–407
5. Hadid A, Pietikäinen M, Ahonen T (2004) A discriminative feature space for detecting and recognizing faces. Proc IEEE Conf Comput Vision Pattern Recogn 2:797–804
6. Jones M, Viola P (2003) Fast multi-view face detection. Mitsubishi Electric Research Lab Technical Report TR-2003-96
7. Levi K, Weiss Y (2004) Learning object detection from a small number of examples: the importance of good features. Proc IEEE Conf Comput Vision Pattern Recogn 53–60
8. Li SZ, Zhang Z (2004) FloatBoost learning and statistical face detection. IEEE Trans Pattern Anal Mach Intell 26:1112–1223
9. Lienhart R, Kuranov A, Pisarevsky V (2002) Empirical analysis of detection cascades of boosted classifiers for rapid object detection. MRL Technical Report. Microprocessor Research Lab, Intel Labs
10. Liu C, Shum H (2003) Kullback-Leibler boosting. Proc IEEE Conf Comput Vision Pattern Recogn 587–594
11. Murphy K, Torralba A, Freeman WT (2003) Using the forest to see the trees: a graphical model relating features, objects, and scenes. Adv Neural Inform Process Syst
12. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24:971–987
13. Porikli FM (2005) Integral histogram: a fast way to extract histograms in cartesian spaces. IEEE Conf Comput Vision Pattern Recogn 829–836
14. Rowley H, Baluja S, Kanade T (1998) Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell 20:23–38
15. Schapire RE, Singer Y (1999) Improved boosting algorithms using confidence-rated predictions. Mach Learn 37:297–336
16. Torralba AB, Murphy KP, Freeman WT (2004) Sharing features: efficient boosting procedures for multiclass object detection. Proc IEEE Conf Comput Vision Pattern Recogn 2:762–769
17. Viola P, Jones MJ (2001) Robust real-time object detection. In: IEEE ICCV workshop on statistical and computational theories of vision
18. Viola P, Jones MJ (57) Robust real-time face detection. Int J Comput Vision 57:137–154
19. Wang H, Li P, Zhang T (2005) Proposal of novel histogram features for face detection. In: Third international conference on advances in pattern recognition. pp 334–343
20. Waring CA, Liu X (2005) Face detection using spectral histograms and SVMs. IEEE Trans Syst Man Cybernet B 35:467–476
21. Wu J, Rehg JM, Mullin MD (2004) Learning a rare event detection cascade by direct feature selection. Adv Neural Inform Process Syst

22. Zhang H, Zhao D (2004) Spatial histogram features for face detection in color images. In: Pacific Rim conference on multimedia. pp 377–384

23. Zhang H, Gao W, Chen X, Zhao D (2006) Object detection using spatial histogram features. Image Vision Comput 24:327–341

24. Zhou J, Lu X, Zhang D, Wu C (20) Orientation analysis for rotated human face detection. Image Vision Comput 20:257–264

25. Zhou S (2005) A binary decision tree implementation of a boosted strong classifier. In: Second international workshop on analysis and modelling of faces and gestures. pp 198–212