Signal Processing 91 (2011) 773–781

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Efficient HOG human detection

Yanwei Pang [a], Yuan Yuan [b,*], Xuelong Li [b], Jing Pan [c]

[a] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, P. R. China
[b] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China
[c] School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin 300222, P. R. China

ARTICLE INFO

Article history:
Received 21 August 2009
Received in revised form
11 July 2010
Accepted 23 August 2010
Available online 16 September 2010

Keywords:
Image and video processing
Human detection
HOG
Fast algorithm

ABSTRACT

While Histograms of Oriented Gradients (HOG) plus Support Vector Machine (SVM) (HOG+SVM) is the most successful human detection algorithm, it is time-consuming. This paper proposes two ways to deal with this problem. One way is to reuse the features in blocks to construct the HOG features for intersecting detection windows. Another way is to utilize sub-cell based interpolation to efficiently compute the HOG features for each block. The combination of the two ways results in significant increase in detecting humans—more than five times better. To evaluate the proposed method, we have established a top-view human database. Experimental results on the top-view database and the well-known INRIA data set have demonstrated the effectiveness and efficiency of the proposed method.

1. Introduction

Object detection is an important step of high-level computer vision. Reliable object detection is essential to image understanding and video analysis. Faces and human bodies are among the most important objects in images and videos. Therefore, face detection and human detection have attracted considerable attention in applications of video surveillance [24,4], biometrics [25], smart rooms, driving assistance systems, social security [2], and event analysis. In this paper, we focus on human detection.

Detecting humans in images is challenging due to the variable appearance, illumination, and background [13,1,4]. Generally speaking, detecting humans in a static image (i.e. a single image or a video frame) is more challenging than in an image sequence. Detecting humans in an image sequence can usually be viewed as moving object detection. So motion information can be used [5,6]. Background can be modeled and used for foreground detection [3,6]. However, in a static image there is no motion clue and the background cannot be modeled. Usually, the core of detecting objects in a static image is effectively modeling the intrinsic characteristics of the objects. This paper limits its scope to human detection in static image.

Feature extraction and classifier designing are two key steps for reliable human detection in a static image. Haar-like rectangle features have been used for detecting humans [15] and faces [16]. To improve the representative and discriminating capacities, the original Haar-like rectangles have been extended to rotated features [17], diagonal features [17], and center-surrounded features [14]. The Haar-like rectangle features encode the intensity contrast between neighboring regions. Such features are suitable for face detection. All frontal faces have similar facial components and the facial components have fixed neighboring relationship. Importantly, the intensity contrast between neighboring facial parts is relatively stable. But the situation in human bodies is quite different. The intensity contrast between regions of a human body depends on the appearance of the humanwear, which varies randomly. So the Haar-like feature is not a discriminating feature for human detection. The human detection performance using merely Haar-like rectangle features is far from acceptable.

[*] Corresponding author.
E-mail address: yuany@opt.ac.cn (Y. Yuan).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.08.010

Scale invariance feature transformation (SIFT) [19] is an alternative feature for human detection [20]. Phung and Bouzerdoum [21] proposed a novel feature called edge-density (ED) for human detection. Another feature taking advantages of edge information is edge orientation histogram (EOH) [22]. Region covariance matrix (RCM) [18,7] is among the state-of-the-art features for human detection. RCM is a matrix of covariance of some image statistics computed inside an image region. It is a matrix-form feature instead of the usual vector-form feature. In this paper we concentrate on the most successful and popular vector-form feature: histograms of oriented gradients (HOG) [9,10,26]. It is inspired by SIFT but different from SIFT. HOG can be regarded as a dense version of SIFT. It is shown that the HOG features concentrate on the contrast of silhouette contours against the background. Finally, it is noted that different types of features can be combined to enhance detection performance [13]. For example, Wang et al. [27] proposed to combine HOG features and local binary pattern (LBP) features in an elegant framework. But feature fusion/combination is beyond the scope of the paper.

The second step of human detection is designing classifier. Large generalization ability and less classifying complexity are two important criteria for selecting classifiers. Linear support vector machine (SVM) [12] and AdaBoost [23] are two widely-used classifiers satisfying the criteria. In this paper, we place emphases on the HOG feature and the SVM classifier and our contributions lie in efficiently computing of HOG features.

Histograms of Oriented Gradients (HOG) plus Support Vector Machine (SVM) [12] is one of the most successful human detection algorithms [9,10,11,27]. The HOG+SVM algorithm [9,10] employs sliding window principle to detect humans in an image. It scans the image at different scales and at each scale examines all the subimages. In each subimage, a 3780-dimensional HOG feature vector is extracted and SVM classifier [12,8] is then used to make a binary decision: human or non-human. Such a detection process is very slow. To overcome the drawback, Zhu et al. [11] proposed to use AdaBoost algorithm to select a small subset of the 3780 HOG features. However, in most cases, the classification accuracy is lower than that of the original HOG+SVM [9,10].

In this paper, we propose to speed up the HOG+SVM algorithm without sacrificing the classification accuracy. The contributions of the papers are: (1) by properly setting the scanning step, the block-based HOG features can be reused for all intersecting detection widows, which significantly reduces the computational cost. (2) We develop a sub-cell based interpolation algorithm to accelerate the calculation of the HOG features in one block, which removes unnecessary (at least unimportant) interpolation while the necessary interpolation is remained; and (3) to deal with occlusion problem; we propose to capture images in top view and we have established a top-view human database.

The rest of the paper is organized as follows: in Section 2, we describe traditional HOG+SVM based human detection algorithm. In Section 3, we describe the proposed method. The experimental results are presented in Section 4. Finally, we provide a brief summary in Section 5.

2. HOG+SVM based human detection

The success of the HOG+SVM human detection algorithm [9,10] lies in its discriminative HOG features and margin-based linear SVM classifier. The HOG+SVM algorithm concentrates on the contrast of silhouette contours against the background [9,10]. Different humans may have different appearances of wears but their contours are similar. Therefore the contours are discriminative for distinguishing humans from non-humans. It is worth noting that the contours are not directly detected. It is the normal vector of the separating hyperplane obtained by SVM that places large weights on the HOG features along the human contours. The HOG+SVM algorithm is outlined as follows:

Input: The scaled input image at the current scale. The size of sliding detection window is 64×128. The sliding step d (d=8 for example).
Output: The locations of the subimages of size 64×128, which are declared to contain humans.
Step 1: For each pixel of the whole image, compute the magnitude $|\nabla f(x,y)|$ and orientation $\theta(x,y)$ of the gradient $\nabla f(x,y)$.
Step 2: From top to bottom and left to right, scan the whole image with a 64×128 window. Extract the 3780 HOG features from the subimage covered by the detection (scanning) window and then apply the leaned SVM classifier on the high-dimensional HOG feature vector to classify the subimage as human or non-human.

The computation of HOG and the principle of SVM are described in the following two sections, respectively.

2.1. HOG

The first step of HOG extraction is to compute the magnitude $|\nabla f(x,y)|$ and orientation (angle) $\theta(x,y)$ of the gradient $\nabla f(x,y)$. The second step of HOG extraction is to derive the orientation histogram from the orientations and magnitudes. The size of the detection window is 64×128, which was experimentally determined for front view humans (see [10,9]). The subimage covered by the detection window (e.g. the dashed rectangles in Fig. 1(a)) is divided into 7×15 overlapping blocks. Each block consists of 4 cells and each cell has 8×8 pixels (see Fig. 1(b)). In each cell the orientation histogram has 9 bins, which correspond to orientations $i \times \pi/9$, i=0,1,…,8 (see Fig. 1(c)). Thus each block contains 4×9=36 features and each 64×128 subimage contains $7 \times 15 \times 36$=3780 features.

Trilinear interpolation can reduce the aliasing effect [9] and therefore is employed to compute HOG features. Trilinear interpolation smoothly distributes the gradient to four cells of a block. In Fig. 2(a), the 4 cells of a block are identified by their centers, (x_i,y_i), i=1,2,3,4. The 4 histograms corresponding to the 4 cells are represented by $h(x_i,y_i,\theta)$ where i=1,2,3,4 and $\theta \in \{0 \times \pi/9, 1 \times \pi/9, \ldots, 8 \times \pi/9\}$. For a given gradient $\nabla f(x,y)$, its orientation $\theta(x,y)$ lies in the range $[i \times \pi/9(i+1) \times \pi/9]$ with i being a proper integer. We describe the range by θ_1=$i \times \pi/9$ and θ_2=$(i+1) \times \pi/9$ (see Fig. 2(b)). Thus $\nabla f(x,y)$ contributes to both $h(x,y,\theta_1)$ and

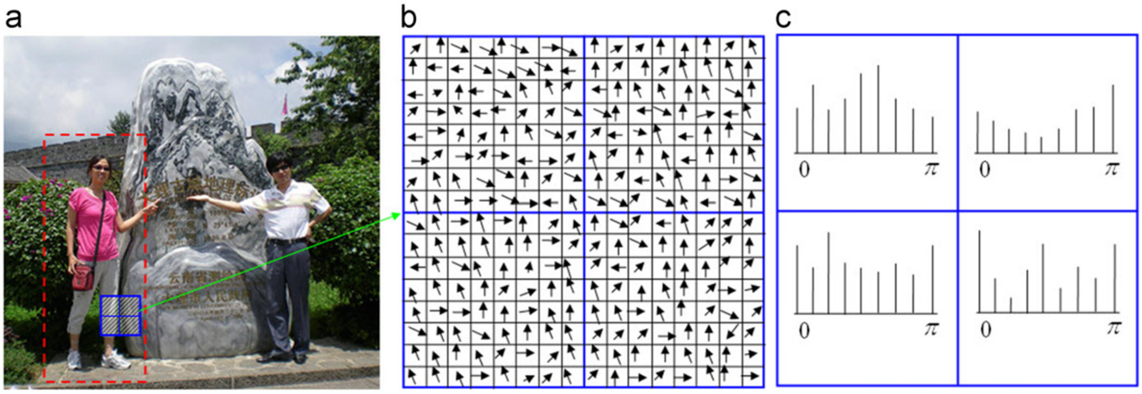

Fig. 1. Histogram of orientation gradients: (a) a 64 × 128 detection window (the biggest rectangle) in an image, (b) a 16 × 16 block consisting of 4 cells, and (c) the histograms of orientation gradients corresponding to the 4 cells.

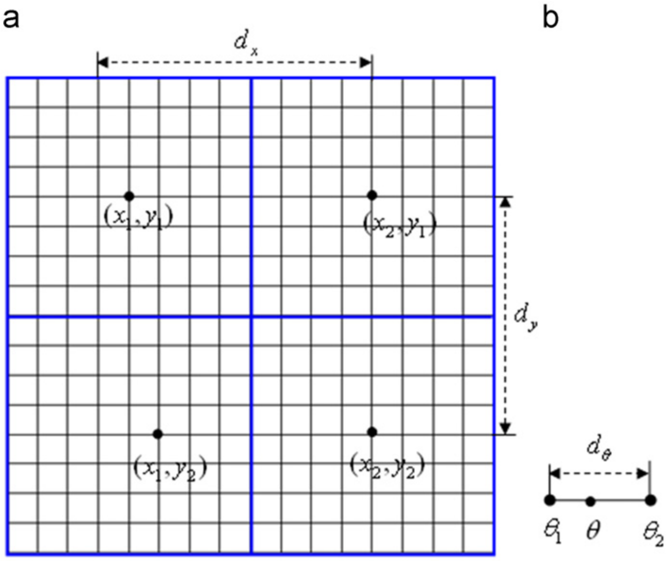

Fig. 2. (a) A block has 4 cells and each cell is identified by its center coordinate, (x_i,y_i), i=1,2,3,4. (b) The gradient at orientation θ contributes to the histograms $h(x,y,\theta_1)$ and $h(x,y,\theta_2)$.

$h(x,y,\theta_2)$. The trilinear interpolation is given by [9]

$$h(x_1,y_1,\theta_1) \leftarrow h(x_1,y_1,\theta_1) + |\nabla f(x,y)|\left(1-\frac{x-x_1}{d_x}\right)\left(1-\frac{y-y_1}{d_y}\right)\left(1-\frac{\theta(x,y)-\theta_1}{d_\theta}\right), \quad (1)$$

$$h(x_1,y_1,\theta_2) \leftarrow h(x_1,y_1,\theta_2) + |\nabla f(x,y)|\left(1-\frac{x-x_1}{d_x}\right)\left(1-\frac{y-y_1}{d_y}\right)\left(\frac{\theta(x,y)-\theta_1}{d_\theta}\right). \quad (2)$$

where d_x=x_2-x_1, d_y=y_2-y_1, and d_θ=$\theta_2-\theta_1$. $h(x_1,y_2,\theta_1)$, $h(x_1,y_2,\theta_2)$, $h(x_2,y_1,\theta_1)$, $h(x_2,y_1,\theta_2)$, $h(x_2,y_2,\theta_1)$ and $h(x_2,y_2,\theta_2)$ can be computed in the similar way.

2.2. SVM

SVM is one of the margin-based classifiers [12]. Given the labeled training data $\{\mathbf{x}_i,y_i\}$, i=1,...,N, $y_i \in \{+1,-1\}$, $x_i \in \mathbb{R}^d$, linear SVM aims at finding an optimal hyperplane $\mathbf{w}^T\mathbf{x}+b=0$, which leads to maximal geometric margin γ [12]:

$$\gamma = \frac{1}{||\mathbf{w}||_2} = \frac{1}{\langle \mathbf{w}\cdot\mathbf{w}\rangle}, \quad (3)$$

where $\langle \cdot \rangle$ stands for inner product between two vectors.

If the training samples are linearly separable, the optimization problem of SVM can be formulated as [12]

$$\min_{\mathbf{w},b} \langle \mathbf{w}\cdot\mathbf{w}\rangle, \\ \text{s.t}\, y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1, \\ i=1,\ldots,N, \quad (4)$$

If the training samples have noise and outliers, they may not be linearly separable. In order to tolerate noise and outliers, slack variables ξ_i are introduced and the corresponding 2-norm soft margin SVM becomes

$$\min_{\mathbf{w},b} \langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{N}\xi_i^2, \\ \text{s.t}\, y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i, \quad i=1,\ldots,N, \quad (5)$$

where C is a free parameter determined by either using a separate validate set or cross-validation technique. Refer to [12] for details about how to solve (5).

3. Proposed algorithm

Two main contributions of the proposed algorithm lie in efficient computation of HOG features: (1) efficient computation of detection-window based HOG features by computing the block-based HOG features in one time and reusing them for all the detection widows intersecting at the block. (2) Efficient computation of block-based HOG features by dividing each cell in a block into 4 sub-cells, classifying them into 3 types, and treating them differently. Thus the cell-based trilinear interpolation is converted into sub-cell based one, which avoids unimportant interpolation. (3) Efficient implementation of pixel based interpolation by the trick of look-up-table (LUT).

3.1. Efficient computation of detection-window based HOG features

It is seen that Eqs. (1) and (2) are applied to each pixel of the 16 × 16 block. So the computation of the 3780-dimensional HOG features in each detection window is time-consuming. Considering that in a scaled image there are a lot of detection windows, the total computational time is therefore very large. Suppose that the image size is 320 × 240 and the scaling factor is 1.1. Then there are totally 2000 detection windows and in each window Eqs. (1) and (2) are applied to compute the 3780-dimensional HOG feature vector. This is the bottleneck for a real-time human detection system.

The proposed algorithm can speed up the feature extraction process. The idea is illustrated in Fig. 3. In Fig. 3(b) the dots are centers of the blocks. The efficiency of the proposed algorithm is achieved by utilizing the fact that: there is a large overlapping area for two neighboring 64 × 128 detection windows (take the two overlapping windows in Fig. 3(a) as an example). As illustrated in Fig. 3(b), if the step between the two neighboring detection windows is proper, the two detection windows intersect with a lot of blocks and the HOG features of these blocks are the same. So independently computing the HOG features in two neighboring detection window is redundant. The proposed algorithm shown below can reduce the redundancy.

Input: The scaled input image at the current scale. The size of sliding detection window is 64 × 128. The sliding step d (d=8 for example). The block has 4 cells and the size of the block is 16 × 16.
Output: The locations of the subimages of size 64 × 128, which are claimed to contain humans.
Step 1: For each pixel of the whole image, compute $|\nabla f(x,y)|$ and $\theta(x,y)$.
Step 2: Image-level process: from top to bottom and left to right, scan the whole image with a block window of size 16 × 16 and step d. Extract the 4 × 9=36-dimensional HOG feature vector from each block.
Step 3: Window-level process: from top to bottom and left to right, scan the whole image with a window of size 64 × 128 and step d. Note that each 64 × 128 detection window covers 7 × 15 blocks and the 36-dimensional HOG feature vector of each block has been obtained in Step 2. Stack (rather than compute) all the 36-dimensional HOG feature vectors in the detection window into a 7 × 15 × 36=3780-dimensional feature vector. Finally, apply leaned SVM classifier on the HOG feature vector to classify the subimage as either human or non-human.

The assumption of the proposed algorithm is that the shift between two neighboring blocks equals the scanning step of the detection windows. The core of the proposed algorithm consists of an image-level process (Step 2) and a window-level process (Step 3). In the image-level process, the 36-dimensonal feature vectors corresponding to the black points (i.e. block centers) in Fig. 3(b) are computed only once and will be used in the window-level process. In the window-level process, all the existing 36-dimensional HOG feature vectors corresponding to a 64 × 128 detection window are concatenated into a

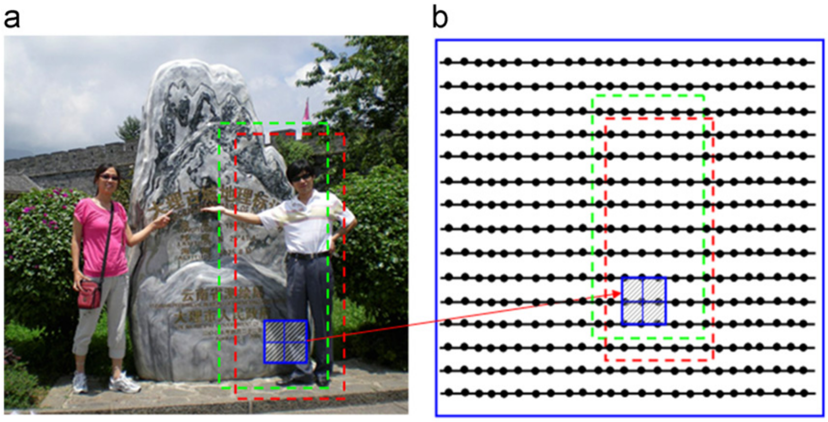

Fig. 3. (a) There is a large overlapping area for two neighboring 64 × 128 detection windows. (b) Most of the blocks of the two detection windows corresponding to (a) are identical.

$7 \times 15 \times 36 = 3780$-dimensional HOG feature vector. The block-based HOG features are obtained only once (image-level process), but can be reused for all the detection widows in the window-level process.

3.2. Sub-cell trilinear interpolation

The above section focuses on computing the HOG features of a detection window, provided that the HOG features in a block are given. This section concentrates on accelerating the calculation of the HOG features in one block.

As stated in Section 2, each block consists of 4 cells. Each HOG feature in one cell is interpolated by not only the gradients in its own cell but also the gradients in other three cells. Smoothly distributing the gradient to 4 cells, this kind of interpolation is beneficial to reduce the aliasing effect [9,10]. Despite its advantage in depressing the aliasing effect, it computational cost is large due to distributing each gradient to all the 4 cells. We call this method as cell-based trilinear interpolation.

To decrease the computation complexity, we develop a sub-cell based trilinear interpolation. The proposed method avoids unimportant interpolation by omitting the gradients whose locations are far from the cell (sub-cell) in concern. Specifically, each cell (8×8 pixels) in a block (16×16 pixels) is divided into 4 sub-cells (there are 4×4 pixels in a sub-cell). We classify the 4 sub-cells into 3 types: corner sub-cell, inner sub-cell, and semi-inner sub-cell. As illustrated in Fig. 4, a block contains 4 cells: C_1, C_2, C_3, and C_4 (see Fig. 4(a)) and C_i contains 4 sub-cells: C_{i1}, C_{i2}, C_{i3}, and C_{i4} (see Fig. 4(b)). Note from Fig. 4(c) that C_{11}, C_{22}, C_{33}, and C_{44} are called corner sub-cells, C_{14},C_{23}, C_{41}, and C_{32} are called inner sub-cells, and C_{12}, C_{21}, C_{24}, C_{42}, C_{43}, C_{34}, C_{31}, and C_{13} are called semi-inner sub-cells. In our algorithm the 3 types of sub-cells have different roles in computing the HOG features:

(1) Because that the corner sub-cells are at the corners of a block and are far from the other 3 cells, the gradients in the corner sub-cells are merely used to compute the histogram corresponding to their own cells. That is, the gradients in C_{11}, C_{22}, C_{33}, and C_{44} contribute only to the histograms of C_1, C_2, C_3, and C_4, respectively.

(2) Because that the inner sub-cells are near to all the 4 cells, so we let the gradients in the inner sub-cells to contribute to the histograms of all the 4 cells.

(3) Each semi-inner sub-cell in a cell is a neighbor of a unique cell. Therefore, the gradients in each semi-inner cell are involved in computing the histograms of its own cell and its neighbor cell. Take the semi-inner cell C_{13} for example, C_{13} is contained in C_1 and is a neighbor of C_3. So the gradients in C_{13} are used for computing the histograms of C_1 and C_3, but they are independent to the histograms of C_2 and C_4.

Let h_1 denote the HOG features of cell C_1. Mathematically loosing, the computation of h_1 is given by

$$h_1 = \sum_{i=1}^{4} hist(C_{1i}) + hist(C_{21}) + hist(C_{31}) + hist(C_{23}) + hist(C_{41}) + hist(C_{32}), \tag{6}$$

where $hist(C_{ij})$ stands for properly interpolated histogram of gradients of the sub-cell C_{ij}. Eq. (6) has 9 items. The first 4 items (i.e. $\sum_{i=1}^{4} hist(C_{1i})$) are associated to all the 4 sub-cells in cell C_1. The second and third items, $hist(C_{21})$ and $hist(C_{31})$, are associated to the semi-inner cells C_{21} and C_{31}, while these last three items correspond to three inner cells, C_{23}, C_{41}, and C_{32}.

Traditional cell-based method can be described by

$$h_1 = \sum_{i=1}^{4} hist(C_i) = \sum_{i=1}^{4} \sum_{j=1}^{4} hist(C_{ij}). \tag{7}$$

In contrast to the proposed sub-cell based method, the traditional method has 16 items while the proposed method has 9 items. Therefore, the computation cost of the proposed method is 9/16=56.25% less than the traditional one in terms of computation of the 3780 HOG features in a block.

3.3. Efficient implementation of pixel based interpolation

Now the question is how to compute the item $hist(C_{ij})$ of Eq. (6). A straightforward way is to utilize Eqs. (1) and (2) to calculate each bin of $hist(C_{ij})$. According to Eqs. (1) and (2), the contribution of gradient at location (x,y) to orientation θ_2 is given by

$$|\nabla f(x,y)| \left(1 - \frac{x - x_1}{d_x}\right)\left(1 - \frac{y - y_1}{d_y}\right)\left(\frac{\theta(x,y) - \theta_1}{d_\theta}\right) = |\nabla f(x,y)| abc, \tag{8}$$

where the coefficients are

$$a = 1 - \frac{x - x_1}{d_x} \tag{9}$$

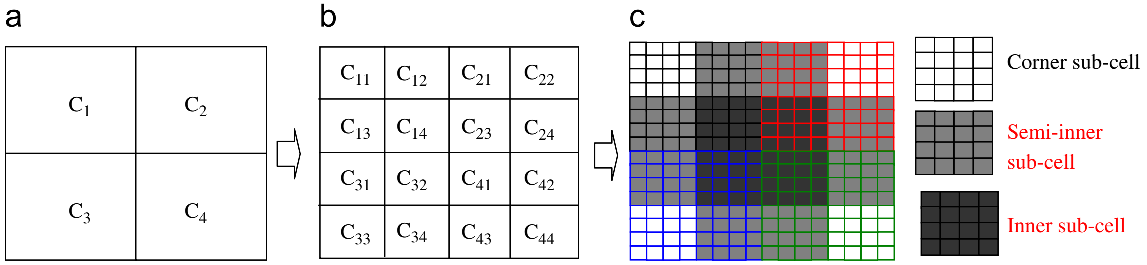

Fig. 4. Sub-cell based representation: (a) a 16×16 block consisting of four 8×8 cells: C_1, C_2, C_3, and C_4. (b) Each 8×8 cell C_i is decomposed into 4 sub-cells: C_{i1}, C_{i2}, C_{i3}, and C_{i4}, which have 3 types: corner sub-cell, inner sub-cell, and semi sub-cell.

$$b = 1 - \frac{y - y_1}{d_y} \tag{10}$$

$$c = \frac{\theta(x,y) - \theta_1}{d_\theta} \tag{11}$$

Generally, it is time-consuming to compute Eq. (8) because it has to calculate the coefficients.

To speed up the computation process, we propose to compute the coefficients a and b offline and save them to a look-up-table. Given (x,y) the coefficients a and b can be obtained by the look-up-table. Note that the coefficient c cannot be computed by the look-up-table trick because the value of $\theta(x,y)$ is continuous instead of discrete. Using the look-up-table, computing Eq. (8) requires only 4 multiplications and 1 addition.

It is worth noting that Wang et al. [27] proposed to estimate Eq. (8) by convolving the gradients at (x,y) with the following 7 by 7 kernel:

$$Conv_{7\times 7} = \frac{1}{256}\begin{bmatrix} 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \end{bmatrix}. \tag{12}$$

Throughout this paper we call this method "convolution method" [27]. The convolution takes 50 multiplications and 49 additions. Even with fast Fourier transformation (FFT) the process takes more operations than our proposed look-up-table method. But the convolution method can be integrated with the trick of integral image [16], so it is faster than the traditional HOG method [10].

4. Experimental results

The proposed algorithms have been evaluated on the INRIA data set [10,9] and our own data set. In Section 3, we have proposed two methods to reduce the cost of computing of the HOG features: one is in detection-window level (Section 3.1) and the other is in block level (see Section 3.2). The detection-window level based HOG computation followed by SVM is called proposed algorithm I (proposed I for simplicity). The proposed method II differs from proposed I in combining detection-window level based and block level based computation of HOG features. We have also compared the proposed method with the traditional method [10,9] and the convolution method (see Section 3.3) [27]. Note that in its original form the convolution method combines HOG features with LBP features and deals with partial occlusion problem. The idea of occlusion processing is inspired by the following interesting phenomenon: the classification score of linear SVM is negative if a portion of the pedestrian is occluded. Based on this observation, the convolution method learns a global detector and a local detector to deal with partial occlusion [27]. However, the occlusion problem is not considered in our paper; the convolution method here does not include the occlusion processing part. In addition, the LBP features are not utilized because the feature fusion is beyond the scope of this paper.

4.1. Results on the INRIA data set

The INRIA data set has been frequently used for developing front-view human detection algorithms. As in [10,9,11], 1208 normalized images were used as positive training samples and 12,180 patches sampled randomly from 1218 person-free training images were used as negative samples. Some of the positive training samples are shown in Fig. 5. The negative samples are randomly sampled from 1218 negative images, the topics of which are about building and outdoor scene. The testing data set includes 288 positive images and 453 negative ones. Fig. 6 shows the instances of the detection results which demonstrate the robustness of the algorithm to variations in illumination, complex background, pose, and partial occlusion.

Furthermore, we adopted recall and precision to measure and compare the performances of the proposed method with traditional HOG+SVM method. Recall is

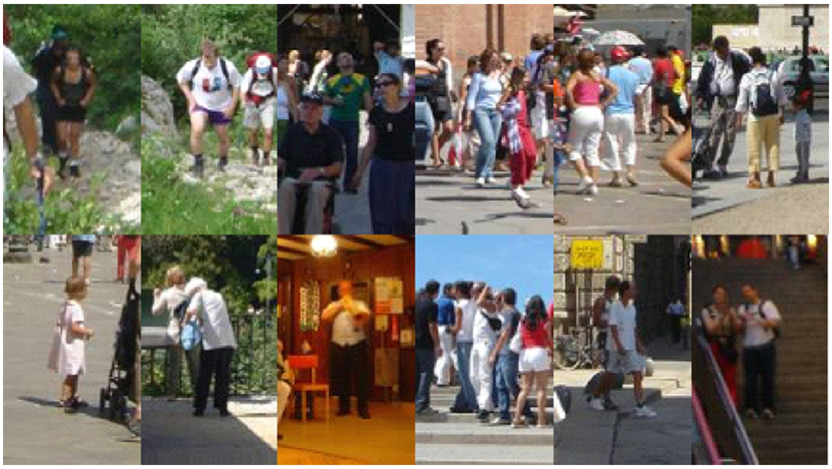

Fig. 5. Twelve examples of positive training images of the INRIA data set. In each image a human is in the central region.

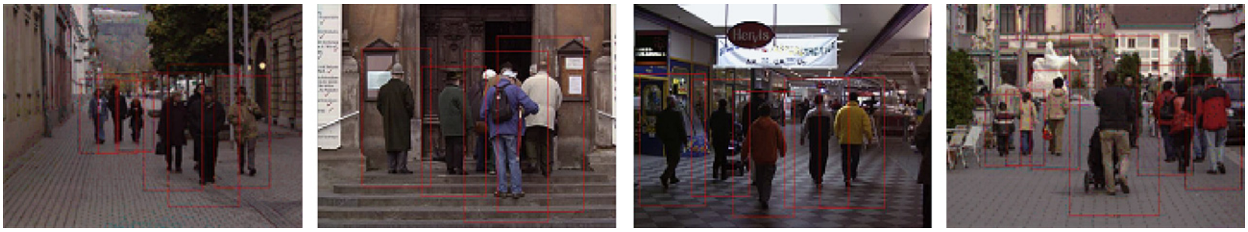

Fig. 6. Examples of detection results on the INRIA data set. Detected humans are enclosed in rectangles.

defined as the number of true positives (i.e. the number of detection windows correctly labeled as belonging to the positive class), *tp*, divided by the total number of detection windows (i.e. the sum of true positives, *tp* and false negatives, *fn*) that actually belong to the positive class:

$$recall = \frac{tp}{tp+fn}. \tag{13}$$

The precision is the number of true positives divided by the total number of detection windows labeled as belonging to the positive class (i.e. the sum of true positives, *tp*, and false positives, *fp*):

$$precision = \frac{tp}{tp+fp}. \tag{14}$$

To deal with overlapping detected windows, we employed the same non-maximum suppression process as the one in [9]. The recall-precision curves are shown in Fig. 7. Because the proposed I has the same performance in terms of recall–precision as traditional HOG+SVM, the recall–precision curve coincides with that of traditional HOG+SVM. From Fig. 7, one can find that the performance of the proposed method II is almost as good as (if not better than) that of the traditional HOG+SVM method, which implies that our sub-cell based computation of the HOG features does not harm the detection performance. When the recall is larger than 0.4, the proposed method is slightly better than the traditional one [10]. In most cases the convolution method is slightly better than the traditional one but not better than proposed method II. As reported in [27], the superiority of the complete convolution method is exhibited when occlusion processing and LBP features are included.

Table 1 shows the detection time. It is observed that the proposed algorithms are roughly five times faster than the original SVM+HOG and is about two times faster than the convolution method. In addition, proposed II is slightly faster than proposed I.

4.2. Results on our data set

From Fig. 6, we can see that severe occlusion occurs between humans because they are captured in front view. To overcome (at least alleviate) the problem, we propose to capture images in top view. With the cameras installed above heads, we have collected our own human data set. The images/videos were collected continuously in fifty days by two cameras. On camera was installed in indoor environment and the other is in semi-outdoor scenario. The normalized positive images are shown in Fig. 8. Negative training images are randomly sampled from the 1074 background images, some of which are shown in Fig. 9. The training data set has 5370 negative

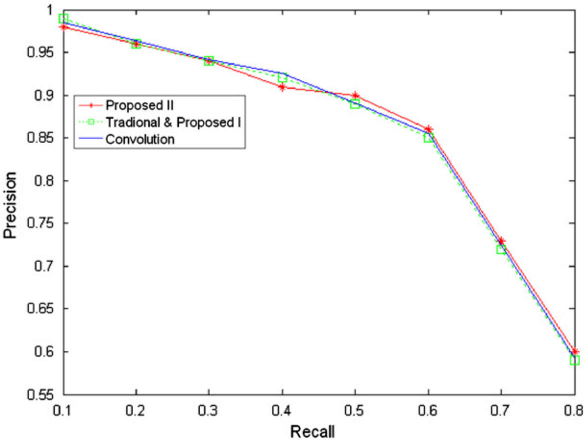

Fig. 7. The recall–precision curves on the INRIA data set. The "convolution" method stands for the method proposed by Wang et al. [27].

Table 1
The detection time (ms) on the INRIA data set.

Scale	1.05	1.10	1.20
Window number	3500	2000	1200
Time (ms)			
HOG+SVM	2250	1285	712
Proposed I	440	215	125
Proposed II	328	159	92
Convolution	752	436	242

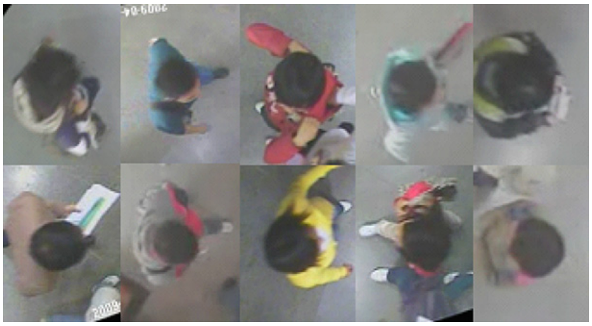

Fig. 8. Ten examples of positive training images of our data set. The images are captured by cameras installed above the heads.

Fig. 9. Four non-human big images. Negative images are randomly sampled from the images.

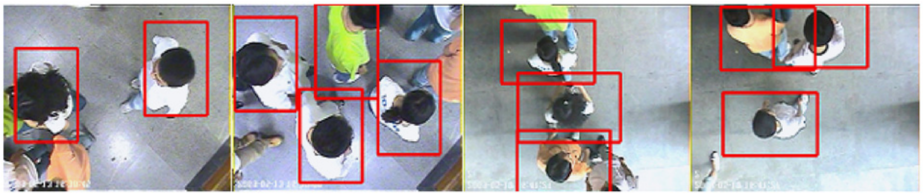

Fig. 10. Examples of detection results on our data set. Detected humans are enclosed in rectangles.

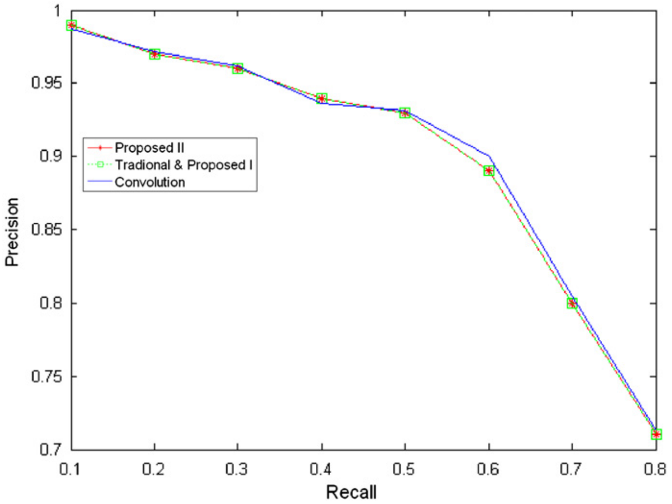

Fig. 11. The recall–precision curves on our data set. The “convolution” method stands for the method proposed by Wang et al. [27].

samples and 2392 positive samples. The testing data set includes 780 positive images and 648 negative ones. The size of the testing images is 320 × 240 pixels. Different from the front-view human detection, the size of the detection window is 80 × 120 and there are 9 × 14 blocks in a detection window.

Fig. 10 shows instances of the detection results which demonstrate the robustness of the algorithm to variations in in-plane rotation, illumination, and complex background. The peoples in the first, second, and third images in Fig. 10 are walking towards bottom left, top right, and top. In each situation, the detector is capable to locate the objects, which shows that the algorithm is insensitive to in-plane rotation. The first two images and the last two images were captured indoors and outdoors, respectively. Though the illumination conditions and backgrounds of the indoor images and outdoor images are quite different, the detection results are satisfying. Fig. 11 shows the detection results (i.e. precision–recall curve) quantitatively. The performance of proposed I is identical to traditional method and proposed II is slightly better than the traditional method. But the convolution method is slightly better than our method.

Table 2
The detection time on our data set.

Scale	1.05	1.10	1.20
Window number	3575	2060	1220
Time (ms)			
HOG+SVM	2300	1323	725
Proposed I	450	221	218
Proposed II	335	164	93
Convolution	767	453	241

Finally, the detection time of the proposed method and the traditional method is shown in Table 2. Similar to Table 1, the proposed methods are roughly five times and two times faster than the traditional method and the convolution method, respectively, which demonstrates the efficiency of the proposed method.

5. Conclusions

We have presented two ways to increase the efficiency of computing the HOG features for human detection. One way is to reuse the features in the blocks to construct the HOG features for a detection window. Another way is to utilize sub-cell based interpolation to efficiently compute the HOG features for each block. Combining the two ways results in more than five times increase in detecting humans in a 320×240 image. The first way significantly reduces computational cost without scarifying any detection accuracy (here recall–precision). The second way slightly (though not significantly) increases the detection accuracy in most cases. In addition, we proposed to efficiently implement the trilinear interpolation by the trick of look-up-table. Finally, we have established a top view human database. Because the view angle is small, the occlusions between humans can be greatly reduced.

Acknowledgements

This work is supported by the National Basic Research Program of China (973 Program) (Grant No. 2011CB707100), the National Natural Science Foundation of China (Grant Nos. 60975001 and 61072093), Tianjin Research Program of Application Foundation and Advanced Technology (Contact No. 10JCYBJC07700), and the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20090032110028).

References

[1] Shufu Xie, Shiguang Shan, Xilin Chen, Xin Meng, Wen Gao, Learned local gabor patterns for face representation and recognition, Signal Processing 89 (12) (2009) 2333–2344.

[2] Bing Xiao, Xinbo Gao, Dacheng Tao, Xuelong Li, A new approach for face recognition by sketches in photos, Signal Processing 89 (8) (2009) 1576–1588.

[3] Dong Xu, Jianzhuang Liu, Xuelong Li, Zhengkai Liu, Xiaoou Tang, Insignificant shadow detection for video segmentation, IEEE Transactions on Circuits Systems for Video Technology 15 (8) (2005) 1058–1064.

[4] Zhong Jin, Zhen Lou, Jingyu Yang, Quansen Sun, Face detection using template matching and skin-color information, Neurocomputing 60 (4–6) (2007) 794–800.

[5] Xiang Ma, F. Bashir, Ashfaq A. Khokhar, Dan Schonfeld, Event analysis based on multiple interactive motion trajectories, IEEE Transactions on Circuits Systems for Video Technology 19 (3) (2009) 397–406.

[6] Shih-Shinh Huang, Li-Chen Fu, Pei-Yung Hsiao, Region-level motion-based foreground segmentation under a Bayesian network, IEEE Transactions on Circuits Systems for Video Technology 6 (4) (2009) 522–532.

[7] Yanwei Pang, Yuan Yuan, Xuelong Li, Gabor-based region covariance matrices for face recognition, IEEE Transactions on Circuits and Systems for Video Technology 18 (7) (2008) 989–993.

[8] Dacheng Tao, Xiaoou Tang, Xuelong Li, Xingdong Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (7) (2006) 1088–1099.

[9] Navneet Dalal, Finding people in images and videos, Ph.D. thesis, INRIA Rhone-Alpes, 2006.

[10] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 886–893.

[11] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1491–1498.

[12] Chistopher J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 121–167.

[13] Yu-Ting Chen, Chu-Song Chen, Fast human detection using a novel boosted cascading structure with meta stages, IEEE Transactions on Image Processing 17 (8) (2008) 1452–1464.

[14] Michael Jones, Paul Viola, Fast multi-view face detection, Tech. Rep. TR2003-096, 2003.

[15] Paul Viola, Michael Jones, Detecting pedestrians using patterns of motion and appearance, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 734–741.

[16] Paul Viola, Michael Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.

[17] Rainer Lienhart, Jochen Maydt, An extended set of Haar-like features for rapid object detection, in: Proceedings of the IEEE International Conference on Image Processing, vol. 1, 2002, pp. 900–903.

[18] Oncel Tuzel, Fatih Porikli, Peter Meer, Pedestrian detection via classification on riemannian manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1713–1727.

[19] David G. Lowe, Distinctive image features for scale-invariant key points, International Journal of Computer Vision 60 (2) (2004) 91–110.

[20] Yan Li, Yanghai Tsin, Yakup Genc, Takeo Kanade, Object detection using 2D spatial ordering constraints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 711–718.

[21] Son Lam Phung, Abdesselam Bouzerdoum, A new image feature for fast detection of people in images, International Journal of Information and Systems Sciences 3 (3) (2007) 383–391.

[22] David Gerónimo, Antonio López, Daniel Ponsa, Angel D. Sappa, Haar wavelets and edge orientation histograms for on-board pedestrian detection, Lecture Notes in Computer Science 4477 (2007) 418–425.

[23] Robert E. Schapire, Yoram Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.

[24] Dacheng Tao, Xuelong Li, Xingdong Wu, and Steve J. Maybank, Human carrying status in visual surveillance, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 1670–1677.

[25] Dacheng Tao, Xuelong Li, Xingdong Wu, Steve J. Maybank, Elapsed time in human gait recognition: a new approach, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2006, pp. 177–180.

[26] Yuan Yuan, Yanwei Pang, Xuelong Li, Footwear for gender recognition, IEEE Transactions on Circuits and Systems for Video Technology 20 (1) (2010) 131–134.

[27] Xiaoyu Wang, Tony X. Han, Shuicheng Yan, An HOG-LBP human detector with partial occlusion handing, in: Proceedings of the IEEE International Conference on Computer Vision, 2009.