IEICE TRANS. INF. & SYST., VOL.E93–D, NO.3 MARCH 2010

LETTER

# Real-Time Human Detection Using Hierarchical HOG Matrices

**Guan PANG**[†a)], ***Student Member***, **Guijin WANG**[†b)], ***Member***, ***and*** **Xinggang LIN**[†c)], ***Nonmember***

**SUMMARY** Human detection has witnessed significant development in recent years. The introduction of cascade structure and integral histogram has greatly improved detection speed. But real-time detection is still only possible for sparse scan of $320 \times 240$ sized images. In this work, we propose a matrix-based structure to reorganize the computation structure of window-scanning detection algorithms, as well as a new pre-processing method called Hierarchical HOG Matrices (HHM) in place of integral histogram. Our speed-up scheme can process $320 \times 240$ sized images by dense scan ($\approx$ 12000 windows per image) at the speed of about 30 fps, while maintaining accuracy comparable to the original HOG + cascade method.

***key words:*** *human detection, real-time detection, HOG, window-scanning, multi-detector*

## 1. Introduction

Human detection is important for many applications in the field of computer vision such as visual surveillance, image retrieval and driver assistance system. Human detection is a challenging task because of the variations in appearance, articulation, posture and illumination condition.

Among various algorithms of human detection, the window-scanning-based type of algorithms is probably the most popular. These algorithms combine local image features into a strong classifier, which is then applied to all possible sub-windows in the input image to detect humans. Viola et al. [3] proposed an algorithm using Haar wavelet features with the Adaboost and cascade training framework [4]. Dalal and Triggs [2] presented a new feature called Histogram of Oriented Gradient (HOG), which is notably more effective for human detection than Haar wavelet.

Recently, the detection accuracy has been further enhanced by new feature types. Wu and Nevatia [5] proposed a silhouette oriented features called edgelet, which was trained for both part detectors and full-body detector to detect inter-occluded humans. Sabzmeydani and Mori [6] described a mid-level feature set called shapelet, trained by two levels of Adaboost. Tuzel et al. [7] suggested utilizing covariance matrices as object descriptors and a learning algorithm on Riemannian manifolds. Lin et al. [8] introduced a multiple instance feature to mitigate the problem of feature misalignment.

Meanwhile, some researchers have attempted to improve the time efficiency of human detection algorithms, which is another important factor for practical applications. Zhu et al. [1] combined HOG feature with the cascade structure, thereby reducing scanning detection time significantly. Wu and Nevatia [9] tried to address the efficiency issue by integrating multiple heterogeneous features with activation priorities. As complex feature like covariance matrix were also included, its time efficiency was still limited.

However, even for the fast HOG and cascade [1] algorithm, real-time detection speed is feasible only for sparse scan on $320 \times 240$ sized images. In this paper, we propose a novel matrix-based feature-scanning structure in place of the traditional window-scanning structure. A new pre-processing method named Hierarchical HOG Matrices (HHM) is also presented to replace integral histogram. We introduce a multi-detector framework to better handle multi-scale human detection. After integrating our propositions with Zhu et al.'s HOG and cascade [1], the detection process triples in speed, achieving 30 frames per second when dense scanning ($\approx$ 12000 windows per image) $320 \times 240$ sized images. In addition, the proposed matrix-based detection structure and HHIM can be further extended to state-of-the-art detection algorithms to achieve both robustness and efficiency.

## 2. The Matrix-Based Detection Structure

First let's revisit traditional window-scanning human detection structure. A scanning window cuts out an image region from the input image to evaluate its probability of containing a human, as shown in Fig. 1. Then the detector computes $n$ different features (e.g. HOG features) at pre-trained location and size consecutively and sends the feature value or vector into corresponding weak classifiers $wc_i$ ($i = 1, \cdots, n$). The outputs of each weak classifier are combined, with voting weights $\alpha_i$, into a strong classifier $sc$, which provides the final evaluation result of the current image region:

$$sc = \sum_{i=1}^{n} \alpha_i \cdot wc_i \tag{1}$$

Finally, the scanning window slides by certain offset to the next image region. The whole process will be repeated until the window has scanned all across the input image.

As the window scan through the input image, we observe that each individual features are also extracted all

Manuscript received October 1, 2009.
Manuscript revised November 17, 2009.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
a) E-mail: pangg07@mails.tsinghua.edu.cn
b) E-mail: wangguijin@tsinghua.edu.cn
c) E-mail: xglin@tsinghua.edu.cn

DOI: 10.1587/transinf.E93.D.658

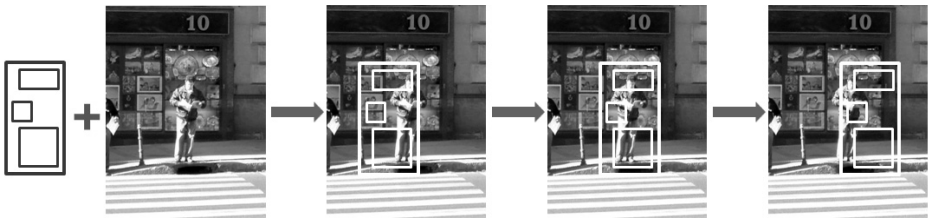

**Fig. 1** In traditional window-scanning detection structure, for each candidate window region, several pre-trained features are evaluated and combined into a final estimation. Then the detection window moves on to the next candidate region.

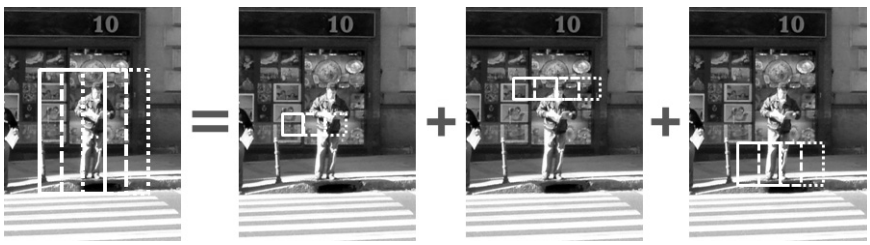

**Fig. 2** Scanning with windows is equivalent to scanning with each feature across the entire input image individually, and adding up the results together to obtain the final estimations for all the candidate windows at once.

across the image, as illustrated in Fig. 2. Thus we propose a new feature scanning style, using a matrix-based computation structure, to replace the traditional window scanning style. For each feature in the ensemble of the strong classifier, the corresponding weak classifier $wc_i$ is first evaluated for the entire image, producing a matrix output $WC_i(x, y)$. Suppose the i-th feature has a relative offset $(x_i, y_i)$ within the detection window. The output matrices $WC_i(x, y)$ are all shifted by this offset as $WC_i(x+x_i, y+y_i)$, so they are aligned to the same origin. The shifted matrices are then weighted by the pre-trained voting weight $\alpha_i$ of each weak classifier, as in Eq. (1). Finally, all matrices are added together into a matrix $SC$:

$$SC(x, y) = \sum_{i=1}^{n} \alpha_i \cdot WC_i(x + x_i, y + y_i) \tag{2}$$

Observe a specific element of matrix $SC$:

$$SC(x_0, y_0) = \sum_{i=1}^{n} \alpha_i \cdot WC_i(x_0 + x_i, y_0 + y_i) \tag{3}$$

Notice that $(x_0 + x_i, y_0 + y_i)$ is the location of the i-th feature for detection window at $(x_0, y_0)$, which means Eq. (3) is equivalent to Eq. (1) for this specific window. Therefore, elements of matrix $SC$ are just the strong classifier outputs, namely the evaluation results, of all candidate window regions.

The benefits of such a matrix-based detection structure are threefold. First, the feature data are processed in the order of data storage in the memory, saving considerable amount of time for data accessing compared to the traditional window-by-window computing order. Second, the detection process is reorganized into matrix operations, enabling the possible utilization of matrix optimizing methods. Finally, the matrix-based structure is suitable for hardware implementation, thereby enhancing the practicality of detection algorithms.

## 3. Hierarchical HOG Matrices (HHM)

To effectively arrange feature data into matrices for the matrix-based detection structure, we propose an alternative to integral histogram in order to pre-process the input image. We name the new pre-processing method as Hierarchical HOG Matrices (HHM).

In our detection framework, the minimum scanning step size is set to 4 pixels, which is dense enough for $320 \times 240$ or larger images. The minimum size of HOG block used is $8 \times 8$, meaning that the minimum size of a single HOG cell is $4 \times 4$ (we use $2 \times 2$ cells for all HOG blocks). As a result, only HOG information at a 4-pixel step size both horizontally and vertically should be recorded during pre-processing. The first step of our pre-processing algorithm is to compute the HOG data for each $4 \times 4$ image patch and store them in a matrix $H_{4\times4}$. $H_{4\times4}$ will then serve as a index for acquiring HOG data of any $4 \times 4$ image patch (at a step size of 4 pixels). Start from $H_{4\times4}$, we can derive matrices for larger HOG size in a hierarchical order, as illustrated in Fig. 3, until we reach the maximum possible HOG size (limited by the size of detection window). Each matrix can be calculated via simple matrix addition from two matrices for smaller HOG size. The results of the whole pre-processing step will be matrices for various HOG size in a hierarchical order, just as indicated by the name Hierarchical HOG Matrices (HHM).

Analysis of computational cost[†] shows that the time re-

[†]Detailed analysis is not presented here due to space limitation. Please contact us if you are interested.

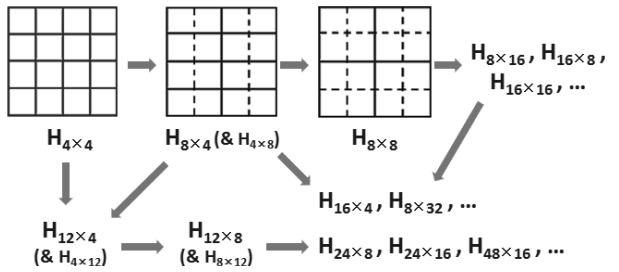


**Fig. 3** The construction of Hierarchical HOG Matrices (HHM): $H_{m\times n}$ denotes the matrix storing HOG data for all $m \times n$ image patches at a step size of 4 pixels. For illustration, suppose the image size is $16 \times 16$, $H_{4\times4}$ would contain $4 \times 4 = 16$ elements. Each element in $H_{8\times4}$ is the sum of two horizontally adjacent elements in $H_{4\times4}$, so it has $3 \times 4 = 12$ elements. Similarly, $H_{8\times8}$ has $3 \times 3 = 9$ elements, and so on.

quired for the construction of HHM is comparable with that of integral histogram. However, integral histogram demands 3 extra addition operations for each HOG value, while HHM does not, because HHM already organized all the useful HOG values for simple data accessing. Considering that thousands of HOG features will be computed for a single image, HHM can lead to considerable speed-up on the detection phase. Note that the tradeoff of the reduced detection time is an increased memory requirement. Experiments show that when pre-processing an $320 \times 240$ image, with all data stored as 32-bit floating point format, integral histogram occupies 3 MB memory, while our HHM need 6.5 MB.

More importantly, HHM arranges the HOG data into matrices according to block size, making it suitable to work in conjunction with the matrix-based detection structure introduced in Sect. 2. The detection process is converted into matrix shift and addition operations, further reducing the detection time by avoiding time wasted on random data accessing.

A potential disadvantage of using HHM is that it's not convenient for multi-scale human detection. As HHM only store data for specific HOG sizes, the calculation of resized HOG blocks (for detection window size other than the default $64 \times 128$) will be difficult. We solved this problem by the introduction of a multi-detector framework. Different detectors are trained for different sizes of detection window. During training, all detectors only select those HOG features contained within HHM. We compute about 30 different HOG sizes during pre-processing to assure a large enough HOG feature pool for selection. This way, in multi-scale detection, each detector will only need to scan with a specifically sized window, without the need to resize it. Values for all encountered HOG features can be fetched directly from HHM. In practice, we only need to train 5 different detectors. Each detector is $1.1 - 1.2$ times larger than the previous detector. Therefore, the 6th detector will be twice as large as the 1st detector and can be obtained by simply resizing the 1st detector. The resized HOG features, which are also twice as large as before, will be included in HHM because of the hierarchical structure. Such multi-detector framework might slow down the training process; but it also enables the usage of HHM, which will greatly accelerate the crucial detection phase.

## 4. Experiments

To evaluate the performance of our proposed algorithm, we implemented it with the support of the Intel OpenCV library using a PC running on a 2.4 GHz Intel CPU. Except for our original matrix-based detection structure and Hierarchical HOG Matrices (HHM), other components of training and detection algorithm are implemented according to what was described in [1] in order to demonstrate the improvement of our algorithm over traditional methods.

We used the INRIA dataset [2] as our training samples. There are 2416 positive training samples in the dataset. For negative samples, we bootstrapped new training samples from the negative training images in the INRIA dataset at the beginning of training each cascade level. As in [1], we chose HOG as feature, linear SVM as the weak classifier, and Adaboost + cascade as the training framework.

Table 1 compares the detection time of our algorithm with several other algorithms, including HOG + SVM by Dalal and Triggs [2], HOG + cascade by Zhu et al. [1], covariance feature by Tuzel et al. [7], integrated feature by Wu and Nevatia [9] and multiple instance feature by Lin et al. [8]. The comparison is carried out in 3 aspects: sparse scan time on a $320 \times 240$ sized image ($\approx$ 800 windows), dense scan time on a $320 \times 240$ sized image ($\approx$ 12000 windows) and windows scanned per second. The comparison demonstrates the time efficiency of our algorithm, especially for dense scan, because dense scan requires evaluation of much more windows and thus computation of much more HOG features. The huge advantage in the number of windows scanned per second also shows the superiority of our algorithm on large-scale window-scanning.

Figure 4 provides some typical detection results of our algorithm. Raw detection results are merged together based on their overlapping ratios. The accuracy of our algorithm is about the same as what was reported by Zhu et al. [1], which means the improvement in detection speed does not lead to any degradation on performance. Theoretically, our new algorithm only reorganizes the computation structure, without any simplification or approximation. Therefore, it will not lead to any difference in performance compared to the algorithm we based on. Our matrix-based detection structure and Hierarchical HOG Matrices (HHM) should be easily extended to any other latest detection algorithms [8], [9] using a window-scanning style and HOG-like features.

## 5. Conclusion

In this paper, we propose a matrix-based detection structure and a new pre-processing method called Hierarchical HOG Matrices (HHM) to replace the traditional window scanning structure and integral histogram. Our goal is to improve the time efficiency of detection algorithms. Experi-

**Table 1** A comparison of detection time on a $320 \times 240$ sized image among various algorithms. For other algorithms, only the data provided by the authors are listed in the table (n/a means no corresponding data). Note the processing time for dense scan.

| | Sparse scan (≈ 800 windows) | Dense scan (≈ 12000 windows) | Windows scanned per second |
|---|---|---|---|
| HOG + SVM [2] | 500 ms | 7000 ms | n/a |
| HOG + cascade (L1 norm) [1] | 26 ms | 106 ms | n/a |
| HOG + cascade (L2 norm) [1] | 30 ms | 250 ms | n/a |
| Covariance feature [7] | n/a | n/a | 3000 |
| Integrated features [9] | n/a | n/a | 24000 |
| Multiple instance feature [8] | 300 ms | n/a | n/a |
| Our algorithm | 21 ms | 33 ms | 800000 |

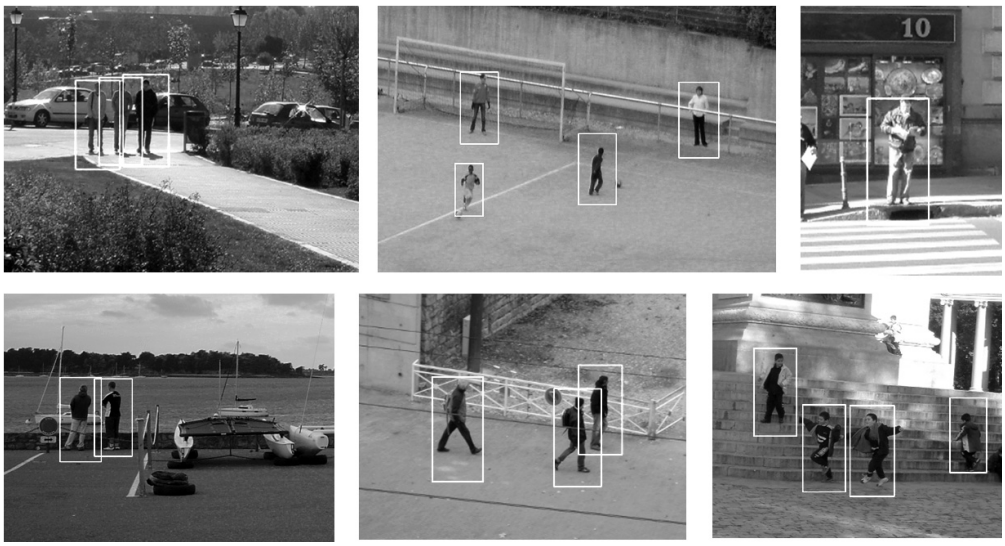


**Fig. 4** Some detection results (after post-processing) of our algorithm.

ments demonstrate that our algorithm can process $320 \times 240$ sized images by dense scan (≈ 12000 windows per image) in real time, while maintaining a competitive performance. It's possible to extend our method to state-of-the-art detection algorithms for speed enhancement.

### References

[1] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," CVPR, vol.2, pp.1491–1498, 2006.
[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," CVPR, vol.1, pp.886–893, 2005.
[3] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using pattern of motion and appearance," ICCV, vol.2, pp.734–741, 2003.
[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," CVPR, vol.1, pp.511–518, 2001.
[5] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," ICCV, vol.1, pp.90–97, 2005.
[6] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," CVPR, vol.1, pp.1–8, 2007.
[7] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifold," CVPR, vol.1, pp.1–8, 2007.
[8] Z. Lin, G. Hua, and L. Davis, "Multiple instance feature for robust part-based object detection," CVPR, vol.1, pp.405–412, 2009.
[9] B. Wu and R. Nevatia, "Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection," CVPR, vol.1, pp.1–8, 2008.