COMPARATIVE EVALUATION OF PEDESTRIAN DETECTION METHODS FOR MOBILE BUS SURVEILLANCE

Wilson S. Leoputra, Svetha Venkatesh, Tele Tan

w.leoputra@postgrad.curtin.edu.au, s.venkatesh@curtin.edu.au, t.tan@curtin.edu.au

ABSTRACT

We present a comparative evaluation of the state-of-art algorithms for detecting pedestrians in low frame rate and low resolution footage acquired by mobile sensors. Four approaches are compared: **a**) The Histogram of Oriented Gradient (HoG) approach [1]; **b**) A new histogram feature that is formed by the weighted sum of both the gradient magnitude and the filter responses from a set of elongated Gaussian filters [2] corresponding to the quantised orientation, called Histogram of Oriented Gradient Banks (HoGB) approach; **c**) The codebook based HoG feature with branch-and-bound (efficient subwindow search) algorithm [3] and; **d**) The codebook based HoGB approach. Results show that the HoG based detector achieves the highest performance in terms of the true positive detection, the HoGB approach has the lowest false positives whilst maintaining a comparable true positive rate to the HoG, and the codebook approaches allow computationally efficient detection.

Index Terms— Pedestrian Detection, Distributed Mobile Sensors, Large Scale Urban Surveillance

1. INTRODUCTION

The pervasive use of CCTV surveillance systems on our public transport vehicles like buses and trains has created demands for new tools to address the large scale spatial and temporal problems in wide area surveillance using multi-source video. Recently, the *VirtualObserver* [4] technology provides a comprehensive approach to index and retrieve, on demand surveillance footage captured from outward facing cameras mounted on buses in urban transport networks. The abilities to locate and subsequently track people of interest using a large number of moving cameras are key problems for law enforcement agencies to deal with crimes on streets. Unlike people detection using static camera footage, the main problem when dealing with moving cameras is that the background scene is not stationary, making standard algorithms like background subtraction [5] unfeasible. Other problems include having to deal with low frame rate and low resolution video as well as environmental variability such as lighting. Crowd levels, variable appearance of pedestrians, and occlusion add to the complexity.

We are motivated by the problem of detecting pedestrians in this real world low frame rate and low resolution footage acquired by a network of mobile cameras mounted on buses and accessed by the Virtual Observer system. Most existing techniques are tested on high frame rate and high resolution video. This work can be classified into two categories: full-body detection [1, 6] and part-body detection [7, 3, 8, 9]. The first approach learns the appearance of the pedestrian as a complete structure and performs the detection in sequential search (sliding window) across the whole image, whilst the latter approach detects a set of local discriminative parts of the pedestrian and aggregates them (spatially or non-spatially) to obtain the final results.

In this paper, we present a comparative evaluation of different pedestrian detection algorithms based on the HoG [1] features on the low resolution and low frame rate video. We apply the Histogram of Oriented Gradient (HoG) approach employed by Dalal and Triggs [1] and evaluate this method against a proposed variant called the Histogram of Oriented Gradient Banks (HoGB), which effectively incorporates multiple scales in its analysis. To overcome the issues of computational complexity in search, we integrate codebooks based on the HoG and HoGB descriptions with the recent branch-and-bound (efficient sub-window search) algorithm [3]. This results in a fast implementation. Our results demonstrate that both with and without the use of codebooks, the incorporation of multiple scales results in lower false positives while maintaining the true positive rate.

This paper is organised as follows. In the following section, we present details of each algorithm for detecting pedestrians. Section 3 describes the datasets and evaluation criteria used in the experiments. Section 4 presents the implementation details and the comparison results of all approaches. Finally, Section 5 concludes the paper.

2. PEDESTRIAN DETECTION

2.1. Histogram of Oriented Gradient (HoG)

The HoG algorithm [1] operates in two steps: 1) building of pedestrian model; 2) pedestrian detection.

Building of pedestrian model: Each detection window is divided into cells of 8×8 pixels. For each cell, HoG computes

the accumulated gradient magnitude corresponds to a quantised orientation to form a 9-bin histogram:

$$\{h(\theta_i) = \sum_{p \in 8\times 8} M_i^p : i = 1, \ldots, 9\}$$

where h denotes the histogram, M_i^p refers to the gradient magnitude corresponding to a quantized orientation θ_i at pixel p, which is computed using 3×3 Sobel kernel. Each patch consists of 2×2 cells. The four histograms in a patch are concatenated to produce a normalised 36-D feature vector. For a detection window of size 96×160 pixels with 7×15 patches, we obtain total of 3780 features. These features are then used to train a linear Support Vector Machine (SVM) classifier.

Pedestrian detection: The detection window is scanned across the entire image to identify the pedestrians. In general, the scanning process has to take into consideration varying scales of the pedestrian, which suffers from high computational cost. To relax this problem, we assume that a pedestrian walks vertically on the ground plane and we employ a foot-to-head calibration strategy to estimate the scale of the pedestrian for a given point in the image. This is done using Homography similar to [10]. Figure 1 shows example of the estimation process. In our application, since the buses travel consistently on the same bus lane on a fixed route, we only need to perform the offline-calibration once for each bus.

2.2. Histogram of Oriented Gradient Banks (HoGB)

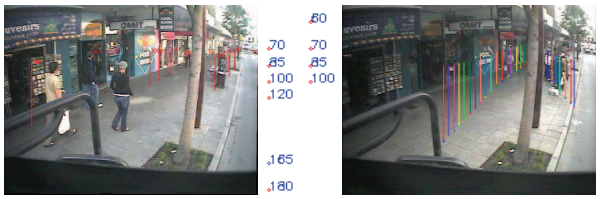
We introduce a new feature for pedestrian detection namely Histogram of Oriented Gradient Banks (HoGB). We further process the gradient image using 9 elongated oriented Gaussian filters [2], and let $\{R_i^p : i = 1, \ldots, 9\}$ be the outputs recorded for each pixel. Figure 2 shows an example of the elongated Gaussian filters (11×11). Then for each cell, the histogram is computed as the weighted sum of the original gradient magnitude and the filter bank response corresponding to the orientation:

$$\{h(\theta_i) = \sum_{p \in 8\times 8} \beta \times M_i^p + (1 - \beta) \times R_i^p\} \qquad (1)$$

where β defines the weight. Intuitively, the first and second term in Equation 1 model the shape of pedestrian at a fine and coarse scale respectively. With the same detection window structure as HoG described in Section 2.1, our final feature vector also consists of 3780 entries per detection window, and we employ similar training strategy to obtain the classifier for detecting the pedestrians.

2.3. Codebook HoG with ESS (CHoG)

Recently, Lampert et al. [3] propose a fast object detection method based on an efficient branch and bound algorithm.



(a) (b) (c)

Fig. 1. Foot-to-head calibration using Homography method. (a) An input image to be calibrated. A set of bars are manually labeled specifying approximated human height. (b) The corresponding 2D point mapping. (c) Given a set of foot positions, the predicted heights are estimated automatically.

Fig. 2. The nine banks of elongated Gaussian filters.

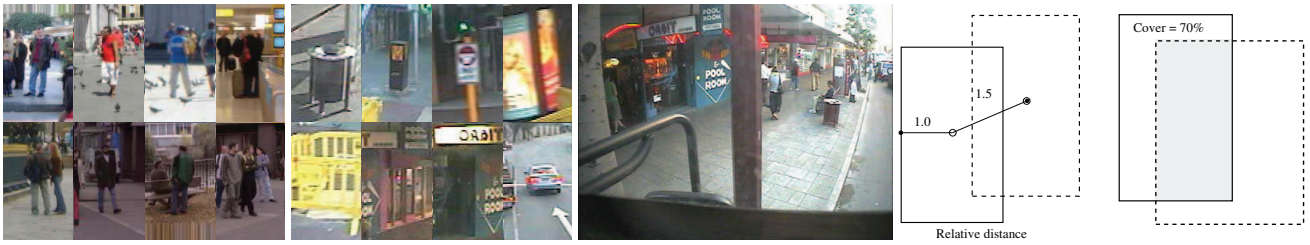
The technique is divided into two steps: 1) creation of codebooks; 2) feature-codebook quantization followed by an efficient search for peak responses. In this paper, we employ a similar idea but use HoG features to build the codebook as an alternative of the Speeded Up Robust Features (SURF) [11].

Building of codebook: The codebook is created by first computing a set of 36-D HoG descriptors from patches of size 20×20 pixels followed by the K-mean clustering. We build two codebooks for positive and negative samples respectively. These codebooks are concatenated and trained using linear SVM classifier to obtain a set of positive and negative alpha (weight), α_i for each codebook entry c_i.

Detection: Given an image, we extract the HoG descriptors and quantize them using the codebook built during offline processing. Thus, each pixel can be represented as $w_i = \sum_i \alpha_i o_i$, where w_i is a score of confidence and o_i is the count of the codebook occurrences. Let the quality function be $f(I) = \sum w_{c_i}$. Hence, we design a function $\hat{f}$ that bounds the values of f over sets of rectangles as:

$$\hat{f}(R) = f^+(R_{max}) + f^-(R_{min}) \qquad (2)$$

where $R = [T, L, B, R]$ is a rectangle defining the [Top, Left, Bottom, and Right] interval coordinates, and each coordinate is defined as $T = [t_{low}, t_{high}]$, $f^+(R_{max})$ represents the positive weight responses for the largest rectangle and $f^-(R_{min})$ is the negative weight responses under the smallest rectangle. Intuitively, Equation 2 always maintain the maximum responses for region R, as it satisfies the bound conditions in [3]. By combining function $\hat{f}$ in Equation 2 with the branch and bound algorithm, we are able to detect pedestrians efficiently.



(a) Positive training images (b) Negative training images (c) Testing image (d) Evaluation criteria

Fig. 3. (a) Examples of the INRIA dataset and (b–c) the Perth dataset that were used for training and testing in our experiments. (d) Evaluation criteria for comparing the ground truth (solid) bounding box with the detected candidate bounding box (dash).

3. DATASETS AND EVALUATION CRITERIA

The two datasets used in our experiments are: the INRIA dataset [12] and the dataset which we collected from 5 buses operated by the Public Transport Authority in Western Australia which we called the Perth dataset. Figure 3 shows examples of the datasets used in our experiments.

Training dataset: We divide the training dataset into positive and negative samples. Each training image is 96×160 pixels in size. For positive training samples, we use the INRIA datasets, which consists of 2416 positives images. Each image contains a person standing against a wide variety of backgrounds including crowds. For negative training samples, we provide a total of 3145 manually cropped background images from the Perth dataset.
Testing dataset: We use the Perth dataset for testing, which was recorded at 7 fps with a resolution of 768×576 pixels. This Perth dataset consists of 1738 frames.
Evaluation criteria: We manually annotate the ground truth for the number of pedestrians in each frame, along with their centroid locations and bounding boxes. There are total of 3521 annotated pedestrians. In this experiment, we are interested in detecting pedestrian having reasonable size (70×145 pixels, ± 30 pixels). In other words, close-up and distance candidates are not included in the ground truth. To evaluate the detection performance, we apply two criteria: *relative distance* and *ratio of the cover* [7], as shown in Figure 3 (d). A detected candidate is considered to be true positive when its *relative distance* from the object is less than 0.5 times the actual size of the ground truth's bounding box and the *cover* is above 50%. Anything else is considered as false positives.

4. EXPERIMENTS

This section presents a comparative evaluation of four pedestrian detectors: HoG, HoGB, codebook HoG, and codebook HoGB. First, we present the implementation details for each detector, including the choice of selecting the parameters and the pre-processing. We present further discussions about the relative performances of these approaches based on detection accuracy and computational speed.

4.1. Implementation details

HoGB: We assess influences of the different scales of the elongated Gaussian filter and the weight (β in Equation 1) on the performance.
◇ SCALE: We evaluate the performance of using elongated Gaussian filters at different scales: (11×11), (15×15), (19×19). We observe that (11×11) scale gives the best performance for the Perth dataset that consists of pedestrians of height ranges between 125-175 pixels.
◇ WEIGHT: Table 1 shows the detection accuracy for varying β in Equation 1. We observe that there is a reduction in both the false alarm and true detection rate when decreasing β (HoG), which implies that by introducing the filter banks, we obtain lower false positives, but sacrifice detection accuracy. Based on the empirical results in Table 1, we choose $\beta = 0.9$.
◇ Similar to the HoG approach, we apply a Gaussian spatial mask and tri-linear interpolation in constructing the HoGB for each patch.

CHoG: The codebook is built as follows:
◇ PRE-PROCESSING: We perform pre-processing on the INRIA pedestrian datasets by selecting patches of size 20×20 pixels around the shape of the pedestrians (shoulders, legs, body, etc) to obtain clean positive samples. We observe an increased detection performance of 5% as a result of this pre-processing step, as the pre-processing helps reduce false quantization during the detection process.
◇ CLUSTERING: As mentioned in [1], the important cues for detecting pedestrian are head, shoulder, leg, and silhouettes. During the creation of codebook, we divide the patches based on its location into two groups: upper and lower parts of the pedestrian (ratio of 30:70); and perform the K-mean clustering independently for each group. This strategy helps improve the final quality of the codebook, since the codebook for the upper body part mainly consists of high curvature features (head and shoulder), whilst the lower part consists of more concentrated straight/vertical features (body and leg).

β	True Positives	False Positives
1.0	0.83	0.37
0.9	0.79	0.31
0.8	0.75	0.29
0.6	0.69	0.26

Table 1. Comparison table of proposed HoGB detectors with various weight parameters for the gradient magnitude and output responses of the elongated Gaussian filters.

	HoG	**HoGB**	**CHoG**	**CHoGB**
True Positive	0.83	0.79	0.67	0.63
False Positive	0.37	0.31	0.40	0.39

Table 2. Performance evaluation of different pedestrian detectors on the Perth dataset.

◇ NUMBER OF CLUSTERS: During the K-mean clustering process, we set $K_u = 100$ and $K_l = 400$ for creating the positive codebook, and $K = 500$ for building the negative codebook, where K_u and K_l are the number of clusters/codebook entries for the upper and lower parts of the pedestrian respectively. Thus, our final codebook consists of 1000 entries.

◇ CHOGB: The CHoGB approach uses the same implementation of CHoG, but using the HoGB feature instead.

4.2. Detection performance

Table 2 shows the comparative performance of the HoG, HoGB, CHoG, and CHoGB. The HoG based detector achieves the highest performance in terms of true positive detection. The HoGB approach has the lowest false positives whilst maintaining a comparable true positive rate compared to the HoG. However, we notice a reduction in true positive rates for the codebook approaches, i.e. CHoG and CHoGB. This is mainly due to the low resolution images and highly cluttered background occurring in the video sequences, leading to false quantization when detecting parts of the pedestrian. Unlike the codebook approach, HoG and HoGB approaches are less sensitive to this problem, as they involve detecting a complete profile of the pedestrian.

4.3. Speed performance

Table 3 shows the speed performances of the four approaches. We see that the CHoG approach is the most computationally efficient of the approaches. It performs 3 times and 4 times faster compared to the original HoG and the HoGB approach respectively. Central to the efficiency is that the codebook approach allows quick search of peak responses on the whole image rather than detection with scanning windows.

	HoG	**HoGB**	**CHoG**	**CHoG**
Overall speed (*in secs*)	45	60	14	19

Table 3. Speed performance for the 4 approaches. Total number of detection window for HoG and HoGB is 6473 per frame.

5. CONCLUSION

In this paper, we present a comparative evaluation of four algorithms for pedestrian detection, along with the implementation details of each approach. In our future work, we will investigate tracking of pedestrians using distributed mobile cameras.

Acknowledgements. This work is based upon work funded in part by DTI and the ARC. We would like to thank Dalal/INRIA research group for their dataset.

6. REFERENCES

[1] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in *ICCV*, INRIA Rhône-Alpes, av. de l'Europe, June 2005, vol. 2, pp. 886–893.

[2] Thomas Leung and Jitendra Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *IJCV*, vol. 43, no. 1, pp. 29–44, June 2001.

[3] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond Sliding Windows: Object Localization by Efficient Subwindow Search," in *CVPR*, June 2008, pp. 1–8.

[4] Stewart Greenhill and Svetha Venkatesh, "Virtual Observers in a Mobile Surveillance System," in *ACM*, New York, NY, USA, 2006, pp. 579–588.

[5] C. Stauffer and W. E. L. Grimson., "Learning Patterns of Activity Using Real-Time Tracking," *PAMI*, vol. 8, no. 22, pp. 747–757, 2000.

[6] Constantine Papageorgiou and Tomaso Poggio, "Trainable Pedestrian Detection," in *ICIP*, 1999, pp. IV:35–39.

[7] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian Detection in Crowded Scenes," in *CVPR*, San Diego, California, USA, June 2005, pp. 878–885.

[8] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," in *CVPR*, Washington, USA, 2006, pp. 1491–1498.

[9] Paul Viola and Michael Jones, "Robust Real-time Object Detection," *IJCV*, 2002.

[10] Zhe Lin and Larry S. Davis and David Doermann, "Hierarchical Part-Template Matching for Human Detection and Segmentation," in *ICCV*, Rio de Janeiro, Brazil, 2007.

[11] Herbert Bay, Tinne Tuytelaars, Van Gool, and L., "SURF: Speeded Up Robust Features," in *ECCV*, Graz Austria, May 2006.

[12] Navneet Dalal, "INRIA Person Dataset," June 2005, http://pascal.inrialpes.fr/data/human/.