OBJECT RECOGNITION SPEED IMPROVEMENT USING BITMAP-HOG

*Alireza Dehghani, David Moloney**

Movidius Plc.
1st Floor, O'Connell Br House,
D'Olier St, Dublin, Ireland.
Email: forename.surname@movidius.com

Ivan Griffin

Emdalo Technologies
Unit 8, Abbey House, Shannon, Ireland.
Email: ivan.griffin@emdalo.com

ABSTRACT

Commonly, HoG/SVM classifier uses rectangular images for HoG feature descriptor extraction and training. This means significant additional work has to be done to process irrelevant pixels belonging to the background surrounding the object of interest. While some objects may indeed be square or rectangular, most of objects are not easily representable by simple geometric shapes. In Bitmap-HoG approach we propose in this paper, the irregular shape of object is represented by a bitmap to avoid processing of extra background pixels. Bitmap, derived from the training dataset, encodes those portions of an image to be used to train a classifier. Experimental results show that not only the proposed algorithm decreases the workload associated with HoG/SVM classifiers by 75% compared to the state-of-the-art, but also it shows an average increase about 5% in recall and a decrease about 2% in precision in comparison with standard HoG.

Index Terms— HoG, object detection, image classification, bitmap

1. INTRODUCTION

Histogram of oriented Gradients (HoG) feature descriptors are used in computer vision for a decade. It was first proposed by Dalal *et al.* [1] for pedestrian detection in static images, although it was expanded later to include human detection in videos, as well as to detect a variety of animals and objects in static imagery. When first introduced, HoG feature descriptors brought about a significant improvement in the state-of-the-art in object detection. This improvement in accuracy was accompanied by a large increase in the computational burden, which limited adoption in embedded solutions particularly.

Ke and Sukthankar [2] reduced the dimensionality of SIFT feature descriptor using the Principal Component Analysis (PCA), whilst they kept it distinctive, robust to image deformations, with faster matching. Inspired by this technique, a number of successful studies [3, 4, 5, 6] reduced the dimensionality of HoG descriptor. Zhu *et al.* [7] integrated a cascade of fast rejectors into HOG, with features chosen from blocks at multiple sizes, locations and aspect ratios. Effective rejector cascades, based on coarse to fine feature hierarchies, were introduced by Zhang *et al.* [8] and Pedersoli *et al.* [9]. These concepts were further refined by Pedersoli *et al.* [10] to accelerate object detection with deformable parts models.

Dollár *et al.* [11] identified the computational bottleneck of many detectors as being the image pyramid. Yamauchi *et al.* [12] focused on reducing memory requirements of HoG descriptors by reducing their dimensionality through generation of binary patterns and by capturing size relationships of HoG features including wild card support in an AdaBoost cascade to improve generalisation. Liu *et al.* [13] contributed a set of "Related HoG" (R-HOG) features. Arróspide and Salgado [14] presented how v-HoG works particularly well for vehicles, and recommend classifier fusion as a promising direction for further progress [15]. Kim and Cho [16] recognised that many of the calculations in vanilla HoG and its overlapping blocks are redundant, and can be diminished without any impact on detection accuracy.

Further to eliminating the runtime efficiency of HoG-like descriptors attained in afore-mentioned works, the current paper introduces a novel approach where the descriptor extraction becomes a two-fold process. The training phase defines the expected shape of the objects' boundaries, then the descriptor extraction uses this prior knowledge to represent the exact topology and extent of object, as advised by the shape-prior. In this way, the background noise of object encoding phase is suppressed while the runtime is eliminated. The rest of this paper is outlined as follow: Section 2 presents motivation. The proposed algorithm, evaluation, and conclusions are discussed in Sections 3, 4 and 5, respectively.

2. MOTIVATION

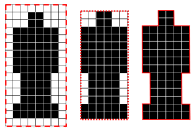
The main application of HoG, when it is used in conjunction with a classifier such as SVM (so-called HoG/SVM), is object detection and classification. Detection, though, involves

*Thanks to Prof. Takeo Kanade for his advises to this work. This work has been partially supported by Project Eyes of Things (EoT) Grant n. 643924 from the European Unions Horizon 2020 Research and Innovation Program.

 The definitive version was published in 2016 XXIII International Conference on Image Processing (ICIP), 25-28 Sept. 2016, pages 659-663, avaliable at IEEE via http://dx.doi.org/10.1109/ICIP.2016.7532439

repeatedly stepping the HoG window across the test image, which manifolds the HoG's computational cost. Moreover, HoG calculation contains no intrinsic sense of the scale and objects can occur at multiple scales in image. Hence, the HoG calculation should be stepped and repeated across each level of a scale pyramid (e.g. 38 levels for 1080p images, the scaling factor of 1.05, and default HoG window). Using the common scaling factor value of 1.05 [1] or 1.2 [17], the test image is repeatedly down-scaled until the scaled image can no longer accommodate a complete HoG window. Clearly, this scaling process intensifies the computational cost of the HoG/SVM. Many optimisations from changing the scale factor to modifying the block-size in which the HoG window is stepped across the scaled image have been suggested to prune the search space and hence limit the computational effort.

The HoG descriptor calculation traditionally implies processing of all foreground and background pixels inside the HoG rectangular window (Fig. 1(a)). The natural extension of this could be limitation of window to the bounding box of object (Fig. 1(b)). Since it eliminates part of the background, it involves less pixels in descriptor computation and thus it alleviates HoG computational cost, reasonably. If this does anything good, even further background elimination, which yields (kind of) sparse bounding contour representation of object (Fig. 1(c)), can bring more amendment in computations. Focusing on only the pixels within this irregular shape is imagined to deliver Less mW, bandwidth, and crucially the dot-product, which is very important for embedded systems.



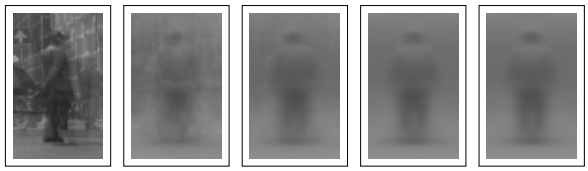
(a) (b) (c)

Fig. 1. The Bitmap Descriptor Hypothesis: (a) Standard HoG window; (b) Hog window limited to the bounding box around10; (c) Bitmap-HoG irregular pattern.

Informed by efficiency of bitmaps in sparse-matrix multiplication, it is used to encode the underlying sparse pattern of object. This improved version of HoG, bitmap-HoG (**bHoG**) hereafter in this paper, improves descriptor's length and thus the computational cost and speed of descriptor computation.

3. BITMAP-HOG (BHOG)

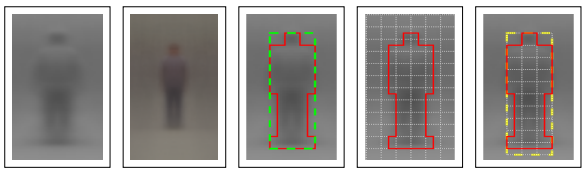
When pedestrian matters, without the lose of generality, the training images of its popular datasets such as INRIA [17] and Daimler [18] contain non-trivial amounts of background in addition to the pedestrian in the foreground. This redundant background area can possibly confound the detector and reduce its effectiveness, apart from the extra computation that it adds. Although $L2$ normalization makes the HoG descriptor background invariant, it cannot solve its computational complexity problem. While, removing the background in bHoG can deliver a significant reduction in descriptor's length and thus the computational costs. This improvement is accepted if bHoG/SVM can deliver at least the same performance as HoG/SVM. To evaluate this, the so-called hypothesized bitmap foreground area should be identified first.



(a) (b) (c) (d) (e)

Fig. 2. Average image of Daimler dataset over image number equal to: (a) 2; (b) 100; (c) 500; (d) 5000; (e) 15660;

The rule-of-thumb idea for estimating the bitmap area is to average the training images of a pedestrian dataset. Apparently, averaging of a few hundred images should be enough to attain the bitmap area. As expected, the average of around 500 training samples of Daimler dataset (Fig. 2(c)) delivers the objective bitmap area. Figs. 3(a) and 3(b) show the average image for the Daimler and INRIA datasets over their whole images, respectively. Clearly, the most relevant information relating to the pedestrians in images of these datasets lies within the red polygon of Fig. 3(c) (the bitmap area).



(a) (b) (c) (d) (e)

Fig. 3. Bitmap area: (a) Daimler dataset average image; (b) INRIA dataset average image; (c) Bitmap area; (d) Bitmap area vs cells; (e) Only the cells inside bitmap area.

3.1. bHoG implementation

bHoG is calculated only for the irregular shape of bitmap rather than the regular-shaped rectangle of standard HoG. The naive approach would be forming a new rectangle image by concatenating the inner cells of bitmap and then calculating the HoG descriptor for this new image. However, this idea creates a different and possibly wrong result. To understand why, it should be noticed that HoG descriptor is calculated for blocks, which are the pre-specified 2×2 (usually) neighbouring cells. Since HoG window is rectangle, all possible blocks of neighbouring cells (starting from top left corner of the window, striding to the right repeatedly toward the bottom right corner) are used in descriptor computation. However, it is not the case for bHoG and only the blocks with the cells inside the bitmap area (*bHoG* ***eligible*** *blocks*) are the mater. Since, the naive idea changes the relative relationship of the cells, the number of blocks as well as the times each cells are involved

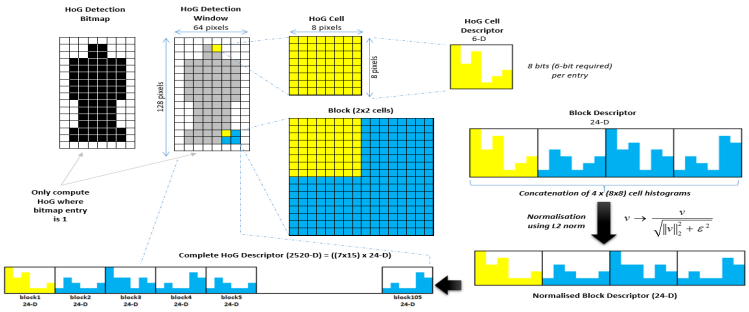


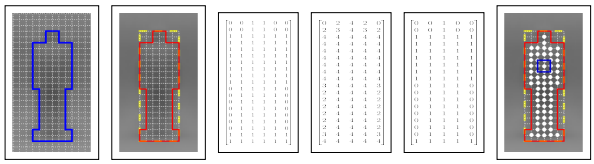
Fig. 4. The Bitmap HoG (bHoG) Descriptor Method.

in bHoG calculation are changed, which leads to wrong result. Accordingly, it is necessary to know the bitmap inner cells, the relative location of the cells regarding the bitmap, and particularly the possible bHoG eligible blocks. Fig. 4 visualizes the way that our hypothesized bitmap idea is applied with respect to the 64×128 pixel HoG window.

On the other hand, decomposing the whole rectangular training image into the cells (as it is in HoG) may make finding the bitmap inner cells complicated (Fig. 3(d)). This complication is due to the cells located on the border of bitmap (so-called *border cells*), which might be outside of bitmap partially. It would be challenging then to decide if these cells are the inner or outer cells. Also, it makes finding the bHoG eligible blocks hard. This would be too severe for higher cell sizes particularly so that bitmap is less probable to have an integer number of inner cells. Obviously, the border cells problem depends on the parameters of HoG and would not happen if the image size is a multiple of cell size (see Fig. 3(a)).

As a remedy, only the bounding box around the bitmap area (the green, dashed box in Fig. 3(c)), can be decomposed into the cells to form the bitmap inner cells. It makes sense because only these cells are the target of bHog and the rest of image is not utilized in the bHoG descriptor calculation and assembly. Moreover, the bounding box might not be a multiple of the cell size depending on the size of cells. Thus, the bounding box should be rounded up or down to compensate for this (e.g. **yellow** bounding box in Fig. 3(e)).

Given the bitmap and HoG parameters (we use Dalal *et al.* [1]'s default parameter in this paper), bHoG descriptor is calculated by concatenating the HoG feature of the eligible blocks. To handle it efficiently, a binary matrix (called *BitMap (BM)* matrix) is defined to identify which cells of the bounding box are inside the bitmap (Fig. 5(c)). Any entry of this matrix is *1* if the corresponding cell belongs to the bitmap, otherwise it would be *0*. In order to determine the eligible blocks, the BM matrix is convolved with a 2×2 kernel of ones matrix. Obviously, size of kernel depends on the number of cells in each block as well as the blocks overlap. The entries of the resultant convolved matrix (Fig. 5(d)) would be between 0 and 4 then. Applying a filtering stage that keep only 4-value entries of BM matrix produces the Contributed Blocks (CB) matrix (Fig. 5(e)) that identify the bHoG eligible blocks. In Fig. 5(f), the eligible blocks are shown by white dots in their centre. The **blue** square in the middle of bitmap is an example eligible block.



(a) (b) (c) (d) (e) (f)

Fig. 5. bHoG computation for cell size = $[4\ 4]$, block size = $[2\ 2]$, and block overlap = $[1\ 1]$: (a) decomposed image; (b) bounding box, bitmap, and cells; (c) BM matrix; (d) convolved BM matrix; (e) CB matrix; (f) eligible blocks.

3.2. Scaling the Bitmap and Bounding Box

As the images of various datasets have differing sizes, a fixed-size bounding box and bitmap would not be general enough to provide the correct bitmap. In order to tackle this, it is necessary to find the location as well as the size of both the bounding box and the associated bitmap area for any new size of image using a kind of scaling factor. If the size of reference image in Fig. 3(c) (Daimler dataset) and the new image (INRIA dataset) are 96×48 and 134×70 respectively, the scaling factor S in x and y direction are calculated as follows:

$$S_x = \frac{70}{48} = 1.458 \quad , \quad S_y = \frac{134}{96} = 1.395 \qquad (1)$$

Despite scaling the bounding box and bitmap, their sizes should be regulated by rounding up or down in order to be an integer multiple of the cell size. Figs. 6(a)-6(d) show the scaling and rounding process versus different cell sizes for an image from INRIA dataset. Fig. 6(d) shows the scaled bounding box in **green** against its rounded up and down versions as **cyan** and **magenta**. Scaling the bounding box scales the bitmap area correspondingly. However, it may displace the vertices of bitmap polygon somewhere inside the cells instead of occurring on the corner of cells. This will cause the previously discussed border cell problem. To avoid this, any vertex of bitmap polygon is rounded to the closest vertex of cell that it falls in. The scaled and rounded bitmap versus cell size are shown in Figs. 6(a)-6(d) in **red** and **magenta**, respectively. Given the scaled and rounded bitmap, the CB matrix is

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 6. (a-d) Scaling versus cell size for a training image of INRIA dataset; (e-h) extracted bHoG features versus the cell size.

used then to extract bHoG descriptor. Figs. 6(e)-6(h) present the extracted bHoG features against the cell sizes.

4. EVALUATIONS

The proposed bHoG algorithm is compared with HoG here.

4.1. Length and Run-time Comparison

To evaluate the improvements that the proposed bHoG approach delivers in terms of the length of descriptor as well as the run-time speed, it is compared with the standard HoG and HoG over the bounding box of bitmap. Although HoG over object's bounding box is not the main objective of this research, it is presented to authenticate the way that bHoG was hypothesized. As Table 1 shows, descriptor length and run-time improvement is started from HoG over bitmap bounding box and subsequently is culminated on bHoG.

Table 1. Length and run-time versus cell size: HoG, HoG over the bounding box of bitmap, and bHoG.

Descriptor	Standard HoG		HoG over bounding box		bHoG	
Cell size	Length	Time (sec)	Length	Time (sec)	Length	Time (sec)
[2 2]	80784	0.312271	32436	0.12712	23184	0.094339
[4 4]	18432	0.083592	7488	0.034892	5220	0.025698
[8 8]	3780	0.025807	1872	0.013434	1116	0.009102
[16 16]	756	0.01336	432	0.008152	180	0.004959

4.2. Precision and Recall Evaluation

A binary pedestrian classification task is composed using the linear SVM method. In this regard, two classes of pedestrian and non-pedestrian are defined for both the training and test sets (Daimler and INRIA datasets, respectively). On this basis, the performance of two HoG/SVM and bHoG/SVM classifiers are compared. In other words, the system is trained using the Daimler dataset and the INRIA images are used for the test. Precision and Recall are considered as the evaluation factors in this regard. As can be seen from Table 2, bHoG delivers approximately the same performance as HoG in terms of precision and recall, i.e., an average increase about 5% in recall and a decrease about 2% in precision in comparison with HoG. Moreover, Fig. 7 compares the RoC curve of bHoG and HoG. As can be seen, bHoG has equivalent performance to classical HoG while cutting memory and computational requirements by up to 4x.

Table 2. Performance comparison between HoG and bHoG for block size = [2 2], block overlap = [1 1].

Cell size	Descriptor	TP	TN	FP	FN	Precision	Recall
[2 2]	*HoG*	951	216	84	175	0.918841	0.844583
	bHoG	994	189	111	132	0.899548	0.882771
[4 4]	*HoG*	943	236	64	183	0.936445	0.837478
	bHoG	1030	204	86	96	0.922939	0.914742
[8 8]	*HoG*	1031	224	76	95	0.931346	0.915631
	bHoG	1084	173	127	42	0.895128	0.9627
[16 16]	*HoG*	1035	193	107	91	0.906305	0.919183
	bHoG	-	-	-	-	-	-

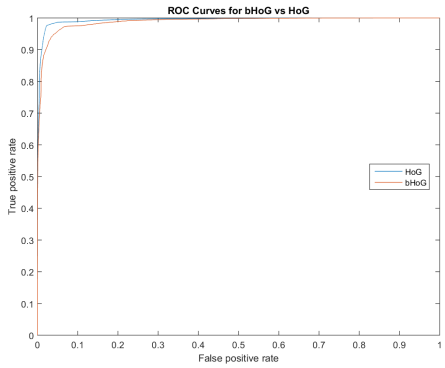


Fig. 7. Comparing the ROC curve of bHoG and HoG.

5. CONCLUSIONS

In this paper, we have proposed an improved HoG approach using the bitmaps to represent the irregular shape of objects. Experimental results on INRIA and Daimler datasets for pedestrian detection, show that the workload associated with HoG/SVM classifiers can be reduced by 75% compared to the state-of-the-art as the bitmap encodes only those pixels or blocks within the reference image useful to the HoG calculation. It has a knock-on effect of reducing the descriptor size required to encode the histogram bins corresponding to the useful pixels/blocks. This performance is gained while there is an average increase about 5% in recall and a decrease about 2% in precision in comparison with the standard HoG. Indeed, the implementation of the detector itself is not optimised allowing significant scope for increasing performance.

6. REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, **1**, pp. 886–893, IEEE, 2005.

[2] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, **2**, pp. II–506, IEEE, 2004.

[3] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *Neural Information Processing*, pp. 598–607, Springer, 2008.

[4] H. Yang, Z. Song, and R. Chen, "An incremental pca-hog descriptor for robust visual hand tracking," in *Advances in Visual Computing*, pp. 687–695, Springer, 2010.

[5] J. Jiang and H. Xiong, "Fast pedestrian detection based on hog-pca and gentle adaboost," in *Computer Science & Service System (CSSS), 2012 International Conference on*, pp. 1819–1822, IEEE, 2012.

[6] A. Dehghani and A. Sutherland, "A dynamic hybrid local-spatial interest point matching algorithm for articulated human body tracking.," in *ICPRAM*, pp. 536–543, 2014.

[7] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, **2**, pp. 1491–1498, IEEE, 2006.

[8] W. Zhang, G. Zelinsky, and D. Samaras, "Real-time accurate object detection using multiple resolutions," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[9] M. Pedersoli, J. Gonzàlez, and J. J. Villanueva, "High-speed human detection using a multiresolution cascade of histograms of oriented gradients," in *Pattern Recognition and Image Analysis*, pp. 48–55, Springer, 2009.

[10] M. Pedersoli, A. Vedaldi, J. Gonzalez, and X. Roca, "A coarse-to-fine approach for fast deformable object detection," *Pattern Recognition* , 2014.

[11] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west.," in *BMVC*, **2**(3), p. 7, Citeseer, 2010.

[12] Y. Yamauchi, C. Matsushima, T. Yamashita, and H. Fujiyoshi, "Relational hog feature with wild-card for object detection," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1785–1792, IEEE, 2011.

[13] H. Liu, T. Xu, X. Wang, and Y. Qian, "Related hog features for human detection using cascaded adaboost and svm classifiers," in *Advances in Multimedia Modeling*, pp. 345–355, Springer, 2013.

[14] J. Arróspide, L. Salgado, and M. Camplani, "Image-based on-road vehicle detection using cost-effective histograms of oriented gradients," *Journal of Visual Communication and Image Representation* **24**(7), pp. 1182–1190, 2013.

[15] J. Arróspide and L. Salgado, "A study of feature combination for vehicle detection based on image processing," *The Scientific World Journal* **2014**, 2014.

[16] S. Kim and K. Cho, "Fast calculation of histogram of oriented gradient feature by removing redundancy in overlapping block," *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING* **30**(6), pp. 1719–1731, 2014.

[17] N. Dalal and B. Triggs, "Inria person dataset," 2005.

[18] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(11), pp. 1863–1868, 2006.