A Refined Object Detection Method Based on HTM

Hongye Liu [1], Taiyin Zhao [2], Yaowei Wang [3], Yonghong Tian [#1]

[1]*School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*
[2]*School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[3]*School of Information and Electronics, Beijing Institude of Technology, Beijing 100081, China*

Emails: {1301214308, yhtian}@pku.edu.cn, tyzhao@uestc.edu.cn, ywwang@jdl.ac.cn

***Abstract*—Object detection plays a fundamental role in many content-based video systems while it is still challenging to preserve a reasonable accuracy and a fairly fast processing speed at the same time. In this paper, we propose a new object detection framework which utilize information from pixel domain and compressed domain jointly. Various pixel-level detection algorithms can be embeded in it and by segmenting the moving region out of the background with the Hit-times Map (HTM), we can save a lot of time since the search area has been restricted to a relatively small region. Once the pixel-level detection finishes, we perform an additional regulation process to refine the coarse detection result employing both temporal consistency and spatial compactness in MV field. The proposed method is tested on a number of standard sequences and achieves a considerable improvement both in detection accuracy and processing speed.**

***Index Terms*—object detection, motion segmentation, compressed domain analysis, video coding, motion vector**

I. Introduction

Object detection is one of the most important components in computer vision and is also the base step of some up-level analysis, such as action recognition, human-computer interaction, surveillance, etc. Although it has been studied for several decades and obtained huge progress in recent years, it remains a challenging work. Many different factors affect the performance of an object detection algorithm and there is no single method that perform good enough in all situations.

Based on which domain the algorithms work in, early works on object detection mainly fall into two categories: **pixel domain methods** and **compressed domain methods**. In general, the former approaches[1–7] will achieve a higher accuracy but also appear to be more computational-demanding. Moreover, most video contents stored in hard disk or transmitted over networks nowadays are not raw pixel data but some kind of encoded bitstreams. Thus, the former approaches require additional computational cost to fully decode the video bitstreams. On the contrary, the compressed domain approaches[8–13] usually process much more faster by utilizing only encoded information such as motion vectors(MVs) and DCT coefficients. Despite the processing speed advantage that the compressed domain algorithms owned, their performance of object detection are often much worse than the pixel domain algorithms. Thus, most compressed domain algorithms are only applicable in some restricted simple scenes.

978-1-4799-6139-9/14/$31.00

Feature representation is very important for object detection and Histograms of Oriented Gradients (HOG)[1] is probably the most popular method in recent years. It is also the basis of a number of other state of the art algorithms including Deformable Parts Model (DPM)[2] and the Exemplar-SVM model[4]. Sparse coding provides another way for feature representation[3, 5, 6]. Ren *et al.* present a method named Histograms of Sparse Codes (HSC)[3] and state it outperforms HOG on some benchmarks like PASCAL and INRIA. Wang *et al.*[7] propose a simple yet effective coding scheme to generate the local feature representation. Moreover, with only linear classifier, this feature coding scheme can achieve a remarkable performance on many benchmarks.

While in the compressed domain, we do not have much information about the visual features in each frame. Thus, we need to use the MVs very carefully. Fei *et al.*[10] use mean shift clustering to segment moving region from background image in H.264 bitstreams. They apply a spatial-range mean shift to find motion-homogenous region and then smoothes the region by temporal-range mean shift. Probabilistic Spatiotemporal Macroblock Filtering (PSMF) and partial decoding are used in [9] to detect and track multiple objects in H.264—AVC Bitstream Domain. There exists another group of methods addressing the detection problem with Markov Random Field (MRF) models. In [12], MVs are quantized into several representative classes and then obtain the MRF label using a maximum a posteriori estimation. The boundaries of segmented moving regions are refined using color and edge information. Khatoonabadi *et al.*[13] propose a new Spatio-temporal Markov Random Field (ST-MRF) model which naturally integrates the spatial and temporal aspects of the objects motion.

This paper aims to provide a new detection framework while preserving high accuracy and fast processing speed. We achieve this by combining the two conventional groups of approaches together. First, a fast moving segmentation operation is performed using the Hit-times Map (HTM) in compressed domain. Second, backing to the pixel domain, we employ a conventional pixel domain detection algorithm to search posible targets in the moving region. Third, a regulation

process combining temporal consistency and spatial compactness of MVs is performed on the coarse detection results. In our experiments, the final HTM-based object detector shows a significant improvement both in detection accuracy and processing speed comparing with the original pixel domain algorithms.

This paper is orginized as follows. Section II presents the details of our proposed method, including how to obtain reliable MVs, how to segment the moving region, etc. Section III demonstrates the evaluating experiments, the corresponding results and some comparisions with other methods. The final conclusions are drawn in Section IV.

II. Proposed Method

In order to preserve a high detection accuracy and obtain a much faster processing speed, we propose to exploit information from pixel domain and compressed domain jointly. Fig.1 shows the framework of our proposed method briefly. In each frame, we get the MVs and raw pixel data from HEVC bitstreams and divide them into two control branch corresponding to the compressed domain and pixel domain respectively. The former branch first approximates MVs of intracoded blocks, removes the unreliable MVs and estimates the moving region as the foreground mask. Then, this foreground mask can be employed to determine the potential object region where the pixel domain detection algorithm will search later. This operation can usually reduce more than half unnecessary computational cost comparing with the original pixel domain detection algorithm which need to search the entire image. After the detection algorithm finishes, there is a post-refinement step to optimize these coarse results.

In our experiments, it is assumed that the background regions of the videos are static and all the target objects are moving all the time. Moreover, the MVs are extracted from HEVC encoded bitstreams. Since our method is based on some existing pixel-level detection algorithms, we choose two of the most popular approaches, HOG[1] and Deformable Parts Model (DPM)[2] to test the improvement in detection accuracy and processing speed with our hybrid framework. Despite the two popular pixel domain detection algorithms and the HEVC coding standard we have chosen in our experiments, some other algorithms and coding standards should also be applicable.

Here we define some notations before going forward. It is assumed that the frame is divided into many small blocks(4x4 in our experiments) and the MV we get from the compressed bitstreams are assigned to each of these blocks. For instance, in frame t, $B_{(x,y)}$ denotes the block located at $(x*b, y*b)$, with its width and height both equal to the block size b ($1 \leq x \leq w/b, 1 \leq y \leq h/b$ and w, h are the width and height of video frame respectively). In this case, the corresponding MV assigned to $B_{(x,y)}$ is $\mathcal{V}_{(x,y)}$, where $\mathcal{V}_{(x,y)} \in \mathbb{R}^2$. Ideally, a block with a zero MV should belong to the background region, which means it doesn't has any change comparing with the reference frame, and the block with a nonzero MV should belong to a moving object.

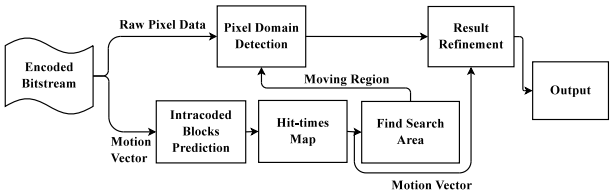


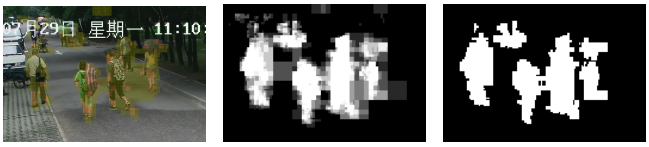
Fig. 1. Flowchart of the proposed method

A. Moving Object Segmentation

For the purpose of reducing the search area of pixel domain object detection algorithms, we need to extract moving region from background first. But it is really dangerous to take the MVs as the foreground region directly since there exists a lot of unreliable MVs among them. After observing a large number of realworld scene sequences, we find that there are mainly three different types of MVs which make it difficult to get the moving region directly: intracoded MVs, random noise and some strangely gathered nonzero MV groups.

Intracoded blocks with no associated MVs may randomly rise up both in foreground and background regions. Here, we employ an effective method called Polar Vector Median(PVM) proposed in [13] to assign proper values to these blocks.

Noise data exists in almost any realworld data and in general, we can elimate them with some filtering methods. But things become a bit different when the strange MV groups appear. They have larger size and sometimes even comparable to the target objects. Thus, the small region filtering methods do not work any more. Fig.2(a) shows the existance of these annoying MVs. Intercoded blocks are labeled with yellow color and intracoded blocks are labeled with orange color. Small region noise MVs have already been elimated. Notice that a number of strange MV groups randomly appear both in foreground region and background region.



(a) (b) (c)

Fig. 2. Hit-times Map. (a) Single frame MVs: intercoded blocks are indicated by yellow color and intracoded blocks are indicated by orange color. (b) HTM of M frames. (c) Foreground mask.

We propose to handle these two types of 'bad' MVs using a new method called Hit-Times Map(HTM) based on a hypothesis that random noise and strange MV groups share together: they seldom rise up continusly in the sequences. In details, we first preserve a buffer of MVs of the past M frames(e.g., $\{\mathcal{V}_{t-M+1}, \mathcal{V}_{t-M+2}, \ldots, \mathcal{V}_t\}$ where t is the current frame index). Then, the occurence times of nonzero MVs can be used to analyse the temporal continuity of each MV block using a simple thresholding operation. We call the occurence of nonzero MVs in M continus frames as Hit-times Map(HTM) and denote it as $\mathcal{H}$. M is the length of a time slice

and should be small enough to prevent strong displacement of target objects. Fig.2 gives a straightforward image of this process: single frame MVs in Fig.2(a) are not suitable to use directly for moving region segmentation while the HTM in Fig.2(b) are much more reliable. The larger $\mathcal{H}_{(x,y)}$ is, the more likely $B_{(x,y)}$ belongs to the moving region. Thus, we can get the foreground mask with a simple thresholding operation:

$$Mask_{(x,y)} = \begin{cases} 1 & \text{if } \mathcal{H}_{(x,y)} \geq \theta_{\mathcal{H}} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\theta_{\mathcal{H}}$ is a constant parameter and we usually assign it with the value of $\frac{2}{3}M$ or $\frac{3}{4}M$.

B. Pixel Domain Object Detection and Result Refinement

Since the pure pixel domain detection approaches often have high computational cost, it will be useful if we can restrict the search region only to the moving region instead of the entire image. And this is straightforward after extracting the moving region mask in the previous step: use a graph connected components search mechanism(e.g. Depth First Search) to find every disjoint region in *Mask* and for each disjoint region, find the smallest rectangles that can cover it. Thus, the search area for pixel domain detection algorithm is restricted to the moving region and we can avoid wasting time searching the entire image.

After the pixel domain detection algorithm finishes, there may still exists a number of false detections. We need to refine each detection result about its confidence score and position. For each detection result $Y^i = (r^i, s^i)$, where $r^i = (r^i_x, r^i_y, r^i_w, r^i_h)$ is the target rectangle and s^i is the confidence score, we have two steps to perform:

1) **Regulate the confidence score.** First, we need to compute the proportion of its nonzero MV blocks defined as

$$p^i = \frac{\#\{(x,y) \mid B_{(x,y)} \in r^i \ \&\ \mathcal{V}_{(x,y)} \neq 0\}}{\#\{(x,y) | B_{(x,y)} \in r^i\}} \quad (2)$$

where $\#$ means the number of elements in the set.
If p^i is extremely small(e.g. 10%), it's reasonable to suspect that r^i is a false detection and then we add a penalty to its confidence score. The penalty function is defined as

$$Penalty(p^i, s^i) = \begin{cases} 0 & \text{if } p^i \geq \theta_p \\ -(C + \lambda e^{-s^i})(\theta_p - p^i) & \text{otherwise} \end{cases} \quad (3)$$

where θ_p is a the nonzero MV proportion threshold.

2) **Regulate the rectangles.** In order to find the similar MV groups, we first convert the MV blocks into a graph model $G = \{V, E\}$:

$$V = \{B_{(x,y)} \mid 1 \leq x \leq \frac{w}{b}, 1 \leq y \leq \frac{h}{b}\} \quad (4)$$

$$E = \{(\mathcal{V}_{(x_1,y_1)}, \mathcal{V}_{(x_2,y_2)}) \mid |x_1 - x_2| \leq 1 \ \& \ |y_1 - y_2| \leq 1 \ \& \ \rho(\mathcal{V}_{(x_1,y_1)}, \mathcal{V}_{(x_2,y_2)}) \leq \theta_s\} \quad (5)$$

where $\rho(v_1, v_2)$ is a similarity function and θ_s is the similarity threshold. $\rho(v_1, v_2)$ is defined as

$$\rho(v_1, v_2) = e^{-\|v_1 - v_2\|^2} \quad (6)$$

Second, search the connected components and the corresponding cover rectangles in this graph, denote as $\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, \ldots, \mathcal{O}^n\}$ and $\mathcal{R} = \{\mathcal{R}^1, \mathcal{R}^2, \ldots, \mathcal{R}^n\}$ respectively($\mathcal{R}^j = (R^j_x, R^j_y, R^j_w, R^j_h)$). Third, compute the proportion of area that each component intersect with r^i

$$q^j = \frac{area(\mathcal{O}^j \cap r^i)}{area(\mathcal{O}^j)} \ (1 \leq j \leq n) \quad (7)$$

The last, update $r^i = (r^i_x, r^i_y, r^i_w, r^i_h)$ as follows:

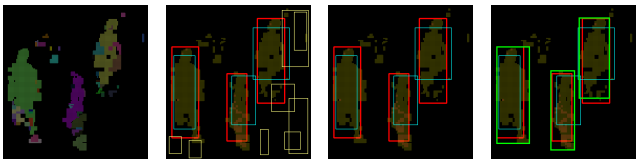
$$r^i_x = \min_j \{R^j_x \mid q^j \geq \theta_q\} \quad (8)$$

$$r^i_y = \min_j \{R^j_y \mid q^j \geq \theta_q\} \quad (9)$$

$$r^i_w = \max_j \{R^j_x + R^j_w \mid q^j \geq \theta_q\} - r^i_x \quad (10)$$

$$r^i_h = \max_j \{R^j_y + R^j_h \mid q^j \geq \theta_q\} - r^i_y \quad (11)$$

Fig.3 shows how these two steps work and the entire refinement process is demonstrated in **Algorithm** 1.



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

Fig. 3. Detection refinement. (a)The connected MV components in different colors. (b-d)The refine process of detection results. Groundtruth is labeled with red color. Rectangles with high nonzero MV proportion are labeled with blue color and yellow rectangles correspond to low nonzero MV proportion. $(b) \rightarrow (c)$ present the score regulation. $(c) \rightarrow (d)$ present the shape regulation. The new detection result is in green color.

Algorithm 1: Refinement of the Detection Result

Input: Original detection result
$Y = \{(r^1, s^1), (r^2, s^2), \ldots, (r^m, s^m)\}$, MVs of the current frame $\mathcal{V}$.
Output: Updated detection result
$Y = \{(r^1, s^1), (r^2, s^2), \ldots, (r^m, s^m)\}$
for $i \leftarrow 1; i \leq m; i \leftarrow i + 1$ **do**
1) $p^i \leftarrow \frac{\#\{(x,y)|B_{(x,y)} \in r^i \ \&\ \mathcal{V}_{(x,y)} \neq 0\}}{\#\{(x,y)|B_{(x,y)} \in r^i\}}$;
if $p^i < \theta_p$ **then**
$s^i \leftarrow s^i + Penalty(p^i, s^i)$;
end
2) MV clustering to get:
$\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, \ldots, \mathcal{O}^n\}$
$\mathcal{R} = \{\mathcal{R}^1, \mathcal{R}^2, \ldots, \mathcal{R}^n\}$
Update r^i using eqs. (8) to (11).
end

III. Experiments

We evaluate the performace of our proposed method on two different datasets:

1) **Pets2012**[1] is a well-established outdoor pedestrian dataset and we choose three sequences in S2.L1 subset for testing: View_001, View_005 and View_006. Each sequence has 795 frames and the resolution of them are 768×576, 720×576 and 720×576 respectively.
2) **PKU-SVD-B**[2] is the second dataset. It is a very challenging realworld dataset. All the sequences are recorded using the full HD(1920×1080) campus surveillance cameras. We evaluate our method on two sequences and each of them has 3000 frames.

On each sequence, we will compare the detection accuracy of original HOG[1], DPM[2] with the new versions embeded in our method(HOG+HTM, DPM+HTM). A predicted bounding box is considered correct if it overlaps more than 40% with a ground-truth bounding box. Fig.4 demonstrates the recall-precision curves of four sequences. It's evident that our method outperforms the original detection approaches a lot both in detection precision and recall. We also record the best F1-scores($F_1 = 2 \times \frac{precision*recall}{precision+recall}$) that each algorithm can achieve on these sequences. The result is presented in Table.I.

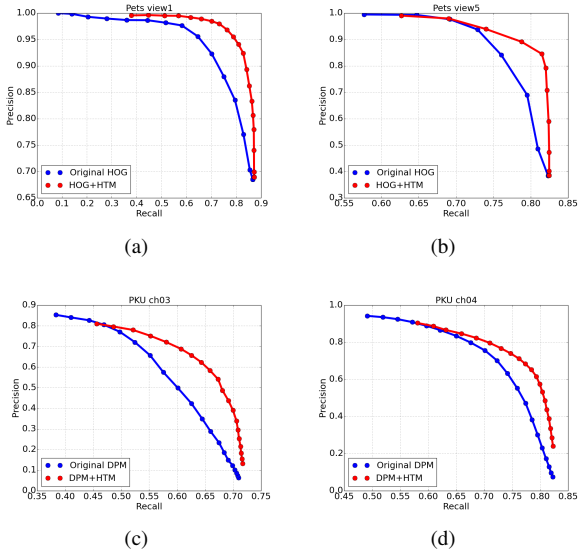


Fig. 4. Recall/Precision curves. (a-b) HOG *vs* HOG+HTM tested on *Pets2012 S2.L1 View1* and *Pets2012 S2.L1 View5*. (c-d) DPM *vs* DPM+HTM tested on *PKU ch03* and *PKU ch04*

IV. Conclusion

In this paper, we present a novel hybrid object detection framework which combines pixel domain detection methods and compressed domain information together. Through effectively utilizing the MVs extracted from compressed video bitstreams, our new object detection method shows not only a faster processing speed but also a higher accuracy comparing with the pure pixel domain approaches. We believe that our work opens up the door for many realtime large-scale surveillance systems which suffer a lot with computational cost and can be extended to more content-based video applications in the future.

[1]website of Pets2012: http://pets2012.net/

[2]website of PKU-SVD-B: http://mlg.idm.pku.edu.cn/resources/dataset.html

TABLE I
The detection F1-score

Sequence		Precision	Recall	F1-score
Pets View1	HOG	0.835	0.795	0.814
	HOG+HTM	0.924	0.828	0.874
	DPM	0.930	0.759	0.836
	DPM+HTM	0.945	0.795	0.864
Pets View5	HOG	0.937	0.728	0.820
	HOG+HTM	0.891	0.788	0.836
	DPM	0.886	0.798	0.840
	DPM+HTM	0.907	0.812	0.857
Pets View6	HOG	0.713	0.695	0.704
	HOG+HTM	0.827	0.651	0.729
	DPM	0.877	0.821	0.848
	DPM+HTM	0.913	0.840	0.875
PKU Ch03	DPM	0.720	0.525	0.607
	DPM+HTM	0.688	0.607	0.645
PKU Ch04	DPM	0.799	0.676	0.732
	DPM+HTM	0.797	0.710	0.751

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China under contract No. 61390515, No. 61035001 and No. 61121002.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.
[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
[3] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *CVPR*. IEEE, 2013, pp. 3246–3253.
[4] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*. IEEE, 2011, pp. 89–96.
[5] M. Aharon, M. Elad, and A. Bruckstein, "-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on SP*, vol. 54, no. 11, pp. 4311–4322, 2006.
[6] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*. IEEE, 2009, pp. 1794–1801.
[7] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*. IEEE, 2010, pp. 3360–3367.
[8] K. Mehmood, M. Mrak, J. Calic, and A. Kondoz, "Object tracking in surveillance videos using compressed domain features from scalable bit-streams," *IEEE Transactions on SP*, vol. 24, no. 10, pp. 814–824, 2009.
[9] W. You, M. H. Sabirin, and M. Kim, "Real-time detection and tracking of multiple objects with partial decoding in h. 264/avc bitstream domain," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 72 440D–72 440D.
[10] W. Fei and S. Zhu, "Mean shift clustering-based moving object segmentation in the h. 264 compressed domain," *IEEE Transactions on IP*, vol. 4, no. 1, pp. 11–18, 2010.
[11] Y.-M. Chen and I. V. Bajic, "A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field," *IEEE Transactions on CSVT*, vol. 21, no. 9, pp. 1316–1328, 2011.
[12] Y.-M. Chen, I. V. Bajic, and P. Saeedi, "Moving region segmentation from compressed video using global motion estimation and markov random fields," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 421–431, 2011.
[13] S. H. Khatoonabadi and I. V. Bajic, "Video object tracking in the compressed domain using spatio-temporal markov random fields," *IEEE Transactions on IP*, vol. 22, no. 1, pp. 300–313, 2013.