HOG-based multi-scale motion detection

Li He[*a], Guijin Wang[a], Qingmin Liao[a]
[a]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

ABSTRACT

Real-time accurate motion detection is a key step for many visual applications, such as object detection, smart video surveillance and so on. Although lots of considerable research efforts have been devoted to it, it is still a challenging task due to illumination variation, etc. In order to enhance the robustness to illumination changes, many block-based motion detection algorithms are proposed. However, these methods usually neglect the influences of different block sizes. Furthermore, they cannot choose background-modeling scale automatically as environment changes. These weaknesses limit algorithm's flexibility and their application scenes. In the paper, we propose a multi-scale motion detection algorithm to benefit from different block sizes. Moreover, an adaptive linear fusion strategy is designed through analyzing the accurateness and robustness of background models at different scales. At detecting, the ratios of different scales would be adjusted as the scene changes. In addition, to reduce the computation cost at each scale, we design an integral image structure for HOG feature of different scales. As a result, all features only need to be computed once. Different out-of-door experiments are tested and demonstrate the performance of proposed model.

Keywords: histogram of gradient HOG; Multi-scale; Linear fusion; Adaptive weights, Integral image

1. INTRODUCTION

Generally, motion detection is a first step in many real-time visual applications, such as objection detection, smart video surveillance, pose recognition and augmented reality. Its accuracy usually effects on the whole performance of the system. Hence, a real-time and accurate motion detection is an important for visual systems.

By now, lots of considerable research efforts have been devoted to motion detection methods and kinds of algorithms have been proposed. In general, there are some necessary requirements to motion detection: it could adapt to the changes of background as time goes and not susceptible to its tiny variations; it should be robust to illumination changes and sensitive to small foreground motion. In fact, robustness and sensitiveness are two opposite sides, many proposed algorithms tried to make a trade-off for the above properties. Most of existing methods can be summarized to two main categories: one is background-model-based method, which firstly establishes a model for the background, and then detects motion by contrasting the difference between image pixel and the learned background model; the other is direct detection methods, which detects foreground motion directly by calculating feature differences in image blocks in image sequence. Comparing with direct detection methods, background-model-based methods can adapt to changes of background, it is robust to tiny variations of background and illumination changes.

For the background-model-based methods, they can be divided into two kinds: one is pixel-based method[1][6]; the other is block-based method. Pixel-based method establishes background models for each pixel in images. When detecting, it calculates the probability of each pixel that it belongs to learned background models. If the probability is smaller than the certain threshold, the corresponding pixel is labeled as foreground, otherwise labeled as background. Due to the multi-models structure, this kind of model could effectively adapt to the variations of leaves in nature scene. However, it is usually susceptible to illumination changes, especially by the sudden large illumination change, such as the flicker of vehicle light in garage. Furthermore, it overlooks the relationships among the nearby pixels, which brings much discrete noise into the final detected motion foreground. Besides, these models need exponential computation which is time-consuming. To the block-based methods, they first partition the whole image into overlapped blocks and then establish models for each block. Since it collects the information among pixels in the relative block, it effectively reduces discrete noise caused by pixels and more robust to illumination changes. But when detecting, all pixels in one block would get the same label, which cause its inaccuracy at motion edges or small objects. Besides, we observed that the sizes of blocks would affect methods' performance. Specifically, the larger sizes of blocks are used, the more robust models against illumination changes could be learned; the smaller sizes are used, the more accurate results could be obtained at the

[*] l-he10@mails.tsinghua.edu.cn; phone +86-13811259347; fax +86-62770317

motion edges or small object. However, to the best of our knowledge, there are no discussions about how to choose sizes of bocks by now.

In this paper, we discuss the effect of blocks with different block sizes and propose a multi-scale motion detection model. Moreover, in order to benefit from the advantages of different blocks, we adopt a linear fusion strategy to adjust weights for each scale model. Besides, to reduce the computation cost at each scale, we design an integral image structure for each channel of HOG feature.

The rest of the paper is organized as follows. In Section 2, we review related works and put forward our motivations. The details of our multi-scale motion detection model are presented in Section 3. At last, we give experiment results and discussion in Section 4 and conclude our work in Section 5.

2. RELATED WORKS

In the field of motion detection, lots of background-model-based methods have been proposed. However, there are still many challenges such as illumination changes, detection of small object, and variation of background and so on.

In 1999, Stauffe et al. proposed GMM (Gaussian Mixture Models)[1] to model the background. It used K weighted Gaussian distributions to model each pixel in feature space. It could efficiently capture the variation of background. However, these models are based the assumption that pixels in the image are independent. In fact, they are likely to be related for some local image region. For example, in the region where leaves shake, pixels belonging to one leaf are dependent. As a result, results of this method usually include much discrete noise. In order to make use of hidden information among pixels, block-based methods[2][3][4] are proposed. In addition, to increase robustness to illumination changes, these methods extract texture feature from images for models. M Heikkila[2][4] utilized LBP (Local Binary Patterns) histograms to establish models for image blocks. And Lei Hu[11] utilized HOG (Histogram of Oriented Gradients) feature[7][12] for models. For these methods, K models would be used for each block to capture the variations of image regions. And in each block, motion could be detected by comparing the difference of two continuous frames. Although the usage of blocks decrease the discrete noise, it cause some other questions: 1) it usually cause an inaccurate result in motion edges due to regarding one block as a whole; 2) optimal block sizes are still difficult; 3) performance of algorithm using texture feature would decrease at flat region. Atsushi Shimada researched the effect of pixel-based method and block-based method to illumination and proposed a fusion model[6] which used and-or grammar to fuse results of GMMs[5] and block-based models[2][4]. Because of simple result fusion strategy, their models cost a great deal computation.

In fact, pixel-based background models can be seen that the size of block-based model is equal to that whose block size is 1×1. Spired by the research[6][9][10] that different block sizes gave different influence on illumination changes, we proposed our motion detection models. Different with their model that fuses pixel-based model and block-based model, we fuse different multi-scale blocks for models. Moreover, we utilize the same data structure for each scale model, which save much computation. Furthermore, not using and-or grammar to fuse the results of different model, we design a self-adapting linear weighted fusion strategy which would adjust weights of each scale block according to current detection results.

3. MULTI-SCALE MODLE

In this section, we present our HOG-based multi-scale model, which consists of basic framework, single scale model, fusion strategy for multiply scales and shared data structure for acceleration.

3.1 Framework

In this subsection, we demonstrate our basic framework of multi-scale model. It consists of N scaled models and their relationships are illustrated by Figure 1. From the framework, it can be seen that we calculate feature only once at the smallest-scale model and for other scales we composite their feature using those of smaller model. For each model, we detect motion independently, and for the final result, we fuse linearly results from every model. When updating parameters, we would adjust models and fusion strategy depending on the results of results of each model and the final result.

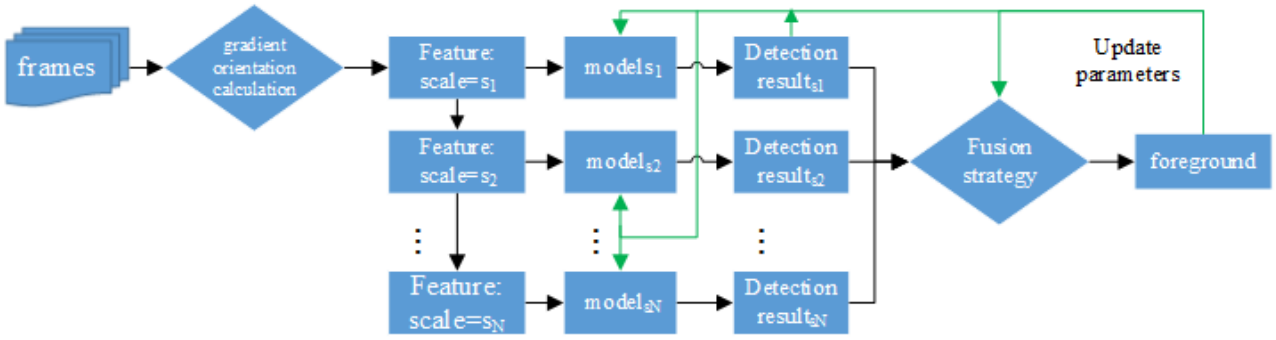

Figure 1. Framework of our method: the black arrows show the flow chart of motion detection; the green arrows show parameters update.

3.2 Single scale model

In general, texture-based model is robust to illumination changes. In our model, we would HOG feature as local description of image block. For each single model, we use the framework proposed by Heikkilä[2][4], but utilize a different updating strategy. We divide image into overlapped blocks as shown in Figure 2. For each block, we extract HOG feature and use K weighted model histograms ($m_{1,t}, m_{2,t}, ..., m_{N,t}$) to model background scene. By multi-models, it can effectively capture variation of background, such as flickering of leaves, flowing stream and so on. We denote the weight of the k^{th} histogram at time instant t by $\omega_{k,t}$. In the following, we discuss how to detect foreground and update background.

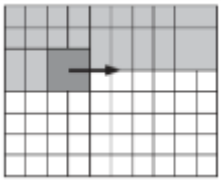

Figure 2. Partially overlapping grid structure used by the algorithm.

We use histogram intersection as the distance measure to compare the new block histogram m_t against learned background model which includes K histograms. The distance for two histograms m_1 and m_2 is defined as follows:

$$dst(m_1, m_2) = \sum_i \min(m_{1,i}, m_{2,i}), \tag{1}$$

where i is the bin index of the histogram. We use a threshold t_b ($t_b \in [0,1]$) to detect foreground. More especially, if the weighted sum of all distances of K learned histograms is larger than t_b , the corresponding block is labeled as foreground, otherwise as background. If none of the model histograms is close enough to the new histogram, the model histogram with the lowest weight is replaced with the new histogram and is given a low initial weight.

If a model histogram close enough to the new histogram is found (their distance is smaller than threshold t_h), the bins of this histogram are updated as follows:

$$m_{k,t}[i] = a_h m_t[i] + (1 - a_h) m_{k,t-1}[i], a_h \in [1,0], \tag{2}$$

where a_h is the updating parameter for model. Different from that in Heikkilä's model[2][4], a_h is constant, we use different values according to the difference of the result of current single-scale model and the final result. Its values are set as:

$$a_h = \begin{cases} v_1, \ if \ \ l_{n,b,t} == l_{b,t} \\ v_2, \ else \end{cases}, (0 \le v_1 < v_2 \le 1), \tag{3}$$

where $l_{n,b,t}$ is the detected result of current single-scale model at time t, $l_{b,t}$ is the final result of multi-scale model at time t. It means that if $l_{n,b,t}$ is consistent with $l_{b,t}$, the learned model histograms would be maintained, updating slowly, otherwise our models would quickly adjust using current scene. Here, using different updating values according to detected results is a different point with Heikkilä's method[2][4]. And the weights $\omega_{k,t}$ are updated as follows:

$$\omega_{k,t} = (1-a_w)\omega_{k,t-1} + a_w 1(dst_k < t_h), a_w \in [1,0], \tag{4}$$

where a_ω is updating parameter. $1(\bullet)$ is an indicator function: if $dst_k < t_h$, $1(\bullet)$ is 1, otherwise 0. Then, K weights should be normalized again so that they sum up to 1.

3.3 Fusion strategy

For the results from different scale models, we use linear fusion strategy to obtain the final result. We denote weight of n^{th} scale model at time t by $\omega s_{n,t}$ and the final result can be obtained as follows:

$$\sum\nolimits_n \omega s_{n,t} * l_{n,b,t} > t_n, t_n \in [0,1], \tag{5}$$

where t_n is a foreground threshold. And the weights $\omega s_{n,t}$ are updated as follows:

$$\omega s_{n,t} = (1-a_n)\omega s_{n,t-1} + a_n 1(l_{n,b,t} == l_{b,t}), a_n \in [1,0], \tag{6}$$

where a_n is the updating parameter. After adjusting all weights, N weights should be normalized again so that they sum up to 1.

3.4 Feature calculation

Integral image is acceleration algorithm for calculating Harr feature proposed by Prikli[13]. In this paper, we can calculate an integral image for each bin of HOG feature as shown in Figure 3(a). Then for any block and any feature bin, it can be calculated in a constant time. Take the shadow block in Figure 3(b) for example, its feature can be calculated: $(A+D)-(B+C)$. More details about integral image can be seen in the paper[13]. By integral image, HOG feature needs to be calculated only once, and the statistic of feature can be done in a constant time for any-size block of any scale model.

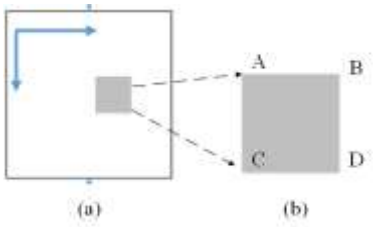

Figure 3. Feature calculation using integral image: (a) is the building of an integral image for one feature bin; (b) is the feature calculation of one feature bin in one block.

4. EXPERIMENTS AND DISCUSSION

In this section, we do experiments in three scenes to demonstrate the performance of our model. They are indoor station platform, outdoor stairs and indoor park. We use three different scales: block sizes are 8×8 pixels, 16×16 pixels and 32×32 pixels. For each scale, we use 50% overlapped block partition and HOG feature to describe background. And other

main parameters are shown in Table 1. The parameter corresponding to dynamic weights F is 10. Our computer configuration is Core2 CPU 1.66GHz, 1.5G and size of image sequence is 320×240. Our processing speed is about 15pbs, which is near real time.

Table 1. Main experiment parameters

scales	a_b	a_w	a_n	t_h	t_b	t_n
8×8	0.01/0.05	0.01	0.05	0.5	0.8	0.8
16×16	0.01/0.05	0.01	0.05	0.6	0.8	0.8
32×32	0.01/0.05	0.01	0.05	0.7	0.8	0.8

To demonstrate the performance of the proposed method, we compare our models with single-scale HOG based model and GMMs [6].

4.1 Comparison with single-scale model

Figure 3 illustrate the results of our proposed method and different single-scale model. We did the comparisons in different scenes (including indoor and our door environments) and we chose three different single-scale models, whose block sizes are 8×8 pixels, 16×16 pixels and 32×32 pixels. The first row images of Figure 3 show the results in the environment that there is no foreground with flickering car light. It can be seen that flickering car light bring much discrete noise, especially in the small-scale model (blocks with 8×8 pixels), while as the scale enlarges, the noise decreases obviously. The right image on the first row is the result of our method. We can see that through fusion strategy, we get clear background. The second row images of Figure 3 are the detected foreground when a car passes with flickering light. In the small-scale model, although foreground (the car) is detected, much noise came along. As scales enlarge, although noise decreases, detected foregrounds become not accurate. In the 32×32 block-size model, we can see that the foreground are is significantly bigger than that of the car. However, our model (the right image of the second row) get more an accurate result than single-scale model with less noise. Similar results also can been obtained in the images on the third and fourth rows. Hence, comparing with single-scale model, our model can get better performance not only in the aspect of noise but also in the aspect of accuracy. Besides, we use the same parameter setting in different scenes, which indicates efficiency of adaptive linear weighted fusion strategy.

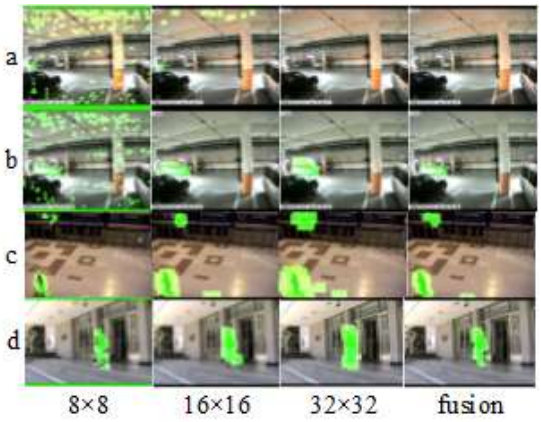

Figure 4. Motion detection results in different scenes: a, no foreground with flickering car light; b, a car drives into the garage with light flickering; c, a man walk through the station platform; d, a man is walking out of door.

4.2 Comparison with GMMs

We compare our method with classical pixel-based model GMMs. The results are illustrated in Figure 4. From results of the first column, we find that the sudden twinkle of car light causes much discrete noise, while our method give much clear background.

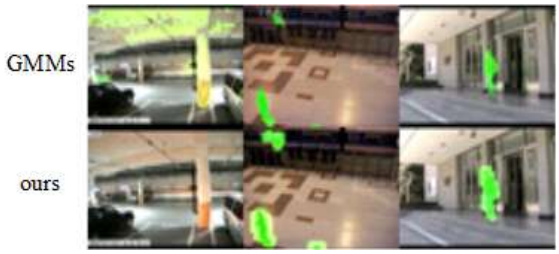

Figure 5. Comparison with GMMs. Images on the first row shows the results of GMMs and those on the second row are our results.

5. CONCLUSION AND FUTURE WORK

In this paper, we analyze the effect of different block-sizes motion detection and propose our HOG-based multi-scale motion detection algorithm, which benefits the robustness to illumination changes of large-scale model and the accuracy of edge region of small-scale model, being proved by experiments in different scenes. As a kind of block-based algorithm, the accuracy of our method at motion edge is not as good as that of pixel-based. Hence, our future work will be focused on how to improve the precision of motion object. Combination of pixel-based method maybe a promising attempt.

6. ACKNOWLEDGEMENTS

This work is partially sponsored by NSFC 61271390.

REFERENCES

[1] Stauffer, Chris, and W. Eric L. Grimson, "Adaptive background mixture models for real-time tracking", Computer Vision and Pattern Recognition, Vol. 2 (1999).

[2] Heikkilä, Marko, Matti Pietikäinen, and Janne Heikkilä, "A texture-based method for detecting moving objects", Proc. British Machine Vision Conf., vol. 1, pp. 187–196 (2004).

[3] Chris Stauffer, W.E.L Grimson, "Adaptive background mixture models for real-time tracking", Computer Vision and Pattern Recognition Vol. 2 (1998).

[4] Heikkilä, Marko, Matti Pietikäinen, "A texture-based method for modeling the background and detecting moving objects", Pattern Analysis and Machine Intelligence, IEEE Transactions on 28.4, pp. 657-662 (2006).

[5] Shimada, Atsushi, Daisaku Arita, and Rin-ichiro Taniguchi, "Dynamic control of adaptive mixture-of-gaussians background model", ideo and Signal Based Surveillance (2006).

[6] Li, Yin, Guijin Wang, and Xinggang Lin, "Three-level GPU accelerated Gaussian mixture model for background subtraction", IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2012.

[7] Guan P, Guijin W, Xinggang L I N, "Real-time human detection using hierarchical hog matrices", IEICE transactions on information and systems, 93(3): 658-661 (2010).

[8] Atsushi Shimada and Rin-ichiro Taniguchi, "Hybrid BackgroundModel using Spatial-Temporal LBP", IEEE, Advanced Video and signal Based Surveillance. pp. 19 – 24 (2009).

[9] Mäenpää, Topi, and Matti Pietikäinen, "Multi-scale binary patterns for texture analysis", Image Analysis. Springer Berlin Heidelberg, pp. 885-892 (2003).

[10] Mäenpää Topi, Pietikäinen Matti, Ojala Timo, "Texture Classification by Multi-Predicate Local Binary Pattern Operators", Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, 3: pp. 951-954 (2000).

[11] Lei Hu, Weibin Liu, Bo Li, "Weiwei Xing, Robust Motion Detection using Histogram of Oriented Gradients for Illumination Variations", ICIMA 2nd International Conference on, pp. 443 – 447 (2010).

[12] Dalal, N., & Triggs, B., "Histograms of oriented gradients for human detection", In Computer Vision and Pattern Recognition, Vol. 1, pp. 886-893 (2005).

[13] Porikli, F., "Integral histogram: A fast way to extract histograms in cartesian spaces", In Computer Vision and Pattern Recognition, Vol. 1, pp. 829-836 (2005).