60 Hz Self-tuning Background Modeling

Jun Luo
School of Automation, Wuhan University of Technology
Wuhan, 430070, China.
junjing2218@gmail.com

Jinqiao Wang
National Laboratory of Pattern Recognition
CASIA, Beijing, 100190, China
jqwang@nlpr.ia.ac.cn

La Zhang
National Laboratory of Pattern Recognition
CASIA, Beijing, 100190, China
zhangla1220@163.com

YingYing Chen
National Laboratory of Pattern Recognition
CASIA, Beijing, 100190, China
yingying.chen@nlpr.ia.ac.cn

Huazhong Xu
School of Automation, Wuhan University of Technology
Wuhan, 430070, China.
wutxhz@163.com

Hanqing Lu
National Laboratory of Pattern Recognition
CASIA, Beijing, 100190, China
luhq@nlpr.ia.ac.cn

ABSTRACT

Background modeling or change detection is often used as a preprocessing step in many computer vision tasks especially for intelligent surveillance. Despite various methods have been proposed to deal with this problem, they often involve complex parameter settings and have poor adaptability to scene changes. In this paper, we propose a fast and robust approach for background modeling with self-adaptive ability. Like ViBe [7], each pixel model is represented by a sequence of historical samples based on sample consensus. To adapt various changes in complex scenes, a flexible feedback scheme is presented to automatically adjust the model parameters. Moreover, a selective diffusion method is employed to overcome the problems like incomplete foregrounds or false detections brought by intermittent moving objects. Experiment results on ChangeDetection benchmark 2014 show that the proposed approach outperforms state-of-the-art approaches with a speed of 60 fps on CPU for a 640×480 image sequence.

CCS Concepts

•**Computing methodologies** → **Scene understanding;**

Keywords

Background modeling, change detection

1. INTRODUCTION

Background modeling or change detection algorithms are used to detect regions of interest (changing or moving areas) and remove background noise in video sequences, which play a significant role in high level surveillance applications, such as object detection, crowd counting, tracking and abnormal detection, etc. Most of state-of-the-art approaches are based on background subtraction, where each frame is matched against the learnt background model to classify pixels into foregrounds and backgrounds. However, most of these methods are sensitive to illumination changes, weather conditions, background/camera motion, shadows, and intermittent moving objects, etc. Moreover, the model parameters need manual setting for different application scenarios.

Background modeling can be approached in many different ways. Gaussian Mixture Modeling(GMM) [4] was a typical representative for pixel based approaches. It assumes that historical color intensities at each pixel can be modeled by a set of Gaussian probability density functions. Kernel Density Estimation (KDE) [1] adopted a non-parametric model based on local intensity observations to estimate background probability density functions at each pixel location. Based on stochastic sampling, ViBe [7] was to model each pixel as a collection of historical observations using a random observation replacement strategy. The codebook modeling method [6] clustered observations into codewords and stored them in local dictionaries, keeping a wider range of representations in the background model.

To automatically adjust model parameters for the background complexity, some feedback mechanisms are presented. Pixel-Based Adaptive Segmenter (PBAS) [5], which used "background dynamics" to adjust thresholds and updating rates, established a feedback scheme to adaptively adjust model parameters. ViBe$^+$ [11] exploited "blinking pixels" to detect dynamic pixels in the current frame to make the feedback loop more robust. SuBSENSE (Self-Balanced SENsitivity SEgmenter) [10] further combined LBSP features and color features with a pixel-level feedback strategy to reduce sensitivity and enhance the generalization capacity. Some model sharing schemes are proposed to effectively exploit the spatial-temporal context for complex scenes, e.g., shared GMM [2] and multi-features based shared models[3].

In this paper, we present a high-efficient and self-tuning background modeling method. Each pixel model is represented by a sequence of historical samples base on sample consensus. Local decision thresholds and update rates of background pixels are adjusted automatically according to the unstable segmentation behaviors and "blinking pixels". Since the initial value of feedback scheme has significant influence on the self-adjusting process, in our method, these initial values of parameters also considered into a feedback

ICIMCS '15, August 19-21, 2015, Zhangjiajie, Hunan, China

DOI: http://dx.doi.org/10.1145/2808492.2808571

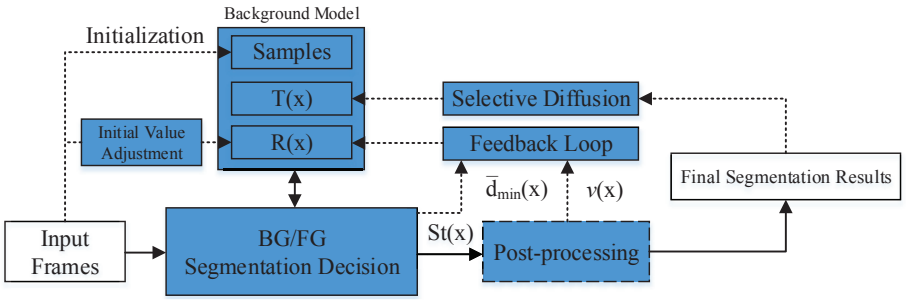

Figure 1: Overview of the proposed approach. Dotted lines are the initialization and the self-tuning process.

loop. Furthermore, we designed a selective-diffusion way to avoid the background updating problems brought by intermittent moving objects.

2. METHODOLOGY

The overview of the proposed approach is illustrated in Figure 1. For an input video sequence, firstly, we build a sample based background model. Then a initial value adjustment is used to adjust the local distance threshold R_x to a appropriate value. Then pixels in input frames are classified into foregrounds and backgrounds by segmentation decision. Finally, the raw segmentation result before post-processing will be used in feedback loop to adjust the local distance threshold R_x and local updating rate T_x, and a selective diffusion rule is designed to adjust diffusion strategy based on the final segmentation result.

2.1 Pixel Modeling with Sample Consensus

Like [7], we adopt a simple background modeling way based on sample consensus. The background model, named B, is formed by a series of pixel models, each of which contains a set of N recent background samples:

$$B(x) = \{B_1(x), B_2(x), ..., B_N(x)\} \tag{1}$$

where N is the number of background samples, which is used to balance the precision and sensitivity of sample consensus. More samples lead to more accurate models but less sensitive for background noise. Additionally, more samples will limit the processing speed. Therefore, in this paper we fixed $N = 35$ to balance accuracy and speed. A pixel x at time t is labelled as foreground 1 or background 0 by matching with $B(x)$ as,

$$S_t(x) = \begin{cases} 1 & if \#\{dist(I_t(x), B_n(x)) < R, \forall n\} < \#min \\ 0 & otherwise \end{cases} \tag{2}$$

where $dist(I_t(x), B_n(x))$ is the distance between pixel x with a given background sample. R is the distance threshold and $\#min$ is the minimum number of matches required for a background classification. We set $\#min$ to 2 as in [7]. The background model B is updated based on the "time subsampling factor" (or the model update rate) T. A randomly selected sample in B has a $1/T$ probability to be replaced by current observation $I_t(x)$.

2.2 Feedback Loop

R and T are two most important parameters in this modeling process. A small R means very accurate segmentation decision but sensitive to the background noise, while a larger R results in better capacity against background disturbances (dynamic background, camera shake, etc) or irrelevant changes (illumination variation, shadow, etc), but makes the segmentation decision harder to detect completely foreground objects when they are very similar to the background. Similarly, "time subsampling factor" T is also hard to choose in these conditions, especially in dealing with intermittent moving objects. A small T will make the slow moving or stationary objects disappeared in the segmentation results, vice versa, a larger T leads to a false detection when the background object moves suddenly. Therefore, a global strategy is difficult to deal with complex scene changes, and the model parameters should be automatically adjusted according to current local situations.

Similar to [5] and [10], we consider R and T as two pixel-level states variables. Two frame-size maps are defined to store the current values of R and T. In feedback loops, they are decided by recursive moving average map $\bar{d}_{\min}$ and blinking pixels accumulators v. The recursive moving average map $\bar{d}_{\min}$ is to measure the background dynamics, which is calculated by the distance between samples and current observations.

$$\bar{d}_{\min}(x) = \bar{d}_{\min}(x)(1-\alpha) + d_{\min}(x) \cdot \alpha \tag{3}$$

where $d_{\min}$ is the minimal normalized distance between samples in $B(x)$ and $I_t(x)$. α is the update rate. Through updating current distance into $\bar{d}_{\min}$, areas with dynamic background would have a high value $\bar{d}_{\min}$. However, when foreground objects stay in the same place for a long time, $\bar{d}_{\min}$ will also reach a high value. To deal with this situation, the frame with blinking pixels F_b is exploited by [11]. F_b is computed by using an XOR operation between current binary segmentation result S_t and previous result S_{t-1},

$$F_b(x) = S_t(x) \otimes S_{t-1}(x) \tag{4}$$

The segmentation results S_t and S_{t-1} here are the raw result without post-processing. Since the borders of moving foreground objects would also be include in F_b, F_b will be filtered by the intersection with the post-processed and dilated version of S_t [10]. The blinking pixel accumulators v is used to calculate variation size of R according to F_b,

$$v(x) = \begin{cases} v(x) + v_{incr} & if\, F_b(x) = 1 \\ v(x) - v_{decr} & otherwise \end{cases} \tag{5}$$

With $d_{\min}$ and v, the distance threshold R_x is adjusted frame by frame,

$$R(x) = \begin{cases} R(x) + v(x) & if\, R(x) > \bar{d}_{\min}(x) \cdot R_{scale} \\ R(x) - \frac{1}{v(x)} & otherwise \end{cases} \tag{6}$$

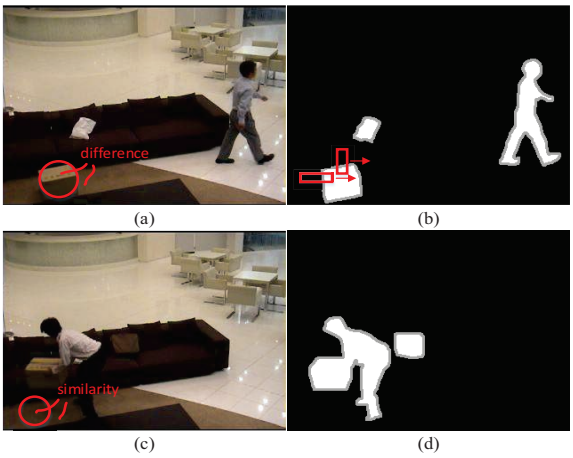

Figure 2: Two different cases for intermittent moving objects. (a) and (c) are input frames, (b) and (d) are the ground truth. The region in red circle in (a) is foreground area of the box, and the region with red circle in (b) is the foreground area which will be falsely detected since the background object (the box) is removed.

where υ_{scale} and R_{scale} are fixed parameters. Similarly, the model update rate is adjusted by:

$$T(x) = \begin{cases} T(x) + \frac{1}{\upsilon(x) \cdot D_{\min}(x)} & if\, S_t(x) = 1 \\ T(x) - \frac{\upsilon(x)}{D_{\min}(x)} & otherwise \end{cases} \quad (7)$$

In the process of a foreground object moving slowly or having a stop, $T(x)$ will increase to keep the object completely. But for the sudden moving of background objects, $T(x)$ increases too, which will lead to false detection results. So a selective diffusion method is designed to deal with this problem section 2.3.

In general, to avoid oscillation in feedback loops, the variation scale of $R(x)$ is restricted to a smaller value conservatively. Therefore, the initial value of $R(x)$ and $T(x)$ has a significant influence on segmentation performance. For instance, scenes with lower light condition needs a small initial value $R(x)$. The automatic adjustment of initial value R_i is decided by the number of foreground pixels in F_b as,

$$R_i = R_i(1 - \frac{L_a}{n+1}) \quad if\, \#\{F_b(x) = 1\} < \#min' \quad (8)$$

where L_a is the average illumination of the current frame. n is the iterations number. In the beginning, R_i has a large initial value, according to light condition of current frame, R_i will decrease to an appropriate value frame by frame until the number of blinking pixels exceeds $\#min'$.

2.3 Selective Diffusion

In ViBe [7], a blind diffusion process was used to update the background model: one sample of random neighbors of $B(x)$ can be replaced by $I_t(x)$ with probability $1/T$, which helped the background model to adapt some scenes like dynamic background or camera jitter. But it cannot deal with the problem brought by intermittent moving objects. ViBe$^+$ [11] proposed a inhibitory propagation rule to solve this condition. But it relies on the gradient on the inner border of background blobs, which limits the processing speed.

As illustrated in Figure 2, the intermittent moving objects can be classified into two categories: one is the moving object staying in a same area for a long time, the other is the background object suddenly moving. Two examples are shown in (a) and (c) in Figure 2. By the analysis of the difference between the local background model and current observations, we apply a selective diffusion scheme to guide the process of model update. In Figure 2(a), the pixels in the foreground area of the box are different from surrounding background pixels. While in Figure 2 (c), since pixels on the box have been updated into the background model, the area behind the box will be falsely classified into foreground pixels for traditional approaches. Contrary to the situation in Figure 2(a), the pixels in the foreground area the box left are similar to surrounding background pixels. So we just need to compute the distance between current observations S_t in foreground area and surrounding background samples in background model $B(x)$, by using decision method in Eq.2. If the distance exceeds a given threshold, we accelerate the diffusion speed of the surrounding pixels, otherwise, we decelerate it. As shown in Figure 2(b), the surrounding areas are detected by computing average gray values in two kinds of sliding windows (window sizes 5×1 and 1×5) in final foreground results (after post-processing, most holes in foreground objects are filled, the pixels in holes will not be treated as surrounding pixels).

With the selective diffusion process, slow moving or sudden stopping objects are kept completely as well as false detection caused by background objects moving can be removed. In addition, another advantage of the selective diffusion is to speed up the ghost removing rate.

3. EXPERIMENTS

To evaluate the performance of the proposed approach, we perform our experiments on the public ChangeDetection benchmark 2014 (CDnet 2014) [12], which provides a realistic, camera-captured, diverse set of videos and contains 53 scenes. To simulate the realistic condition in visual surveillance, we fixed initial parameters of background models in whole experiment process.

3.1 Comparison with the State-of-the-art

Table 2 shows the comparison results between our approach with several state-of-the-art approaches. The results of compared approaches are obtained in the implementation of BGSLibrary [8]. Since we fixed initial value, there is slightly difference with the results reported in CDnet 2014. Our approach achieves the best overall performance and the best individual performance in five of eleven categories, especially in the category "Intermittent Object Motion", our method exceed the second result over 10%. For the self-tuning of the initial value, our approach achieves best performance in scenes with low light conditions like "Night Video" and "Thermal". Furthermore, without any optimization in the c/c++ implementation, our approach runs 60 frames per second for 640×480 pixels on Intel i7 CPU 3.4 GHz.

4. CONCLUSIONS

In this paper, we present a high-efficient and self-tuning background modeling approach. With the dynamic background information and blinking pixels, all the parameters are automatically adjusted through feedback loops, includ-

Table 1: F-measures for subset of CDnet 2014 benchmark [12]. BW: Bad Weather; Ba: Baseline; CJ: Camera Jitter; DB: Dynamic Background; IOM: Intermittent Object Motion; LF: Low Framerate; NV: Night Video; Sh: Shadow; Th: Thermal; Tu: Turbulence; Overall is the average F-measure of 11 categories.

Approach	Ba	BW	CJ	DB	IOM	LF	NV	PTZ	Sh	Th	Tu	Overall
SuBSENSE[9]	**0.9490**	**0.8551**	**0.7752**	0.8146	0.5967	0.6254	0.4811	**0.3839**	**0.8995**	0.6827	**0.8684**	0.7211
ViBe[7]	0.8604	0.6967	0.5703	0.4845	0.4965	0.3714	0.3765	0.06241	0.7973	0.6477	0.4520	0.5287
GMM1[4]	0.4520	0.3593	0.2902	0.2474	0.2387	0.4773	0.3647	0.2457	0.4361	0.2982	0.2304	0.3309
GMM2[14]	0.5198	0.2481	0.4270	0.3622	0.2140	0.6490	0.3502	0.2470	0.3924	0.2322	0.1092	0.3410
KDE[1]	0.6756	0.4563	0.6102	0.2097	0.3817	0.4460	0.1703	0.0345	0.6097	0.3996	0.5066	0.4091
MutltiLayerBGS[13]	0.7450	0.4177	0.6728	0.5672	0.3166	0.5895	0.4253	0.3251	0.7434	0.3138	0.6510	0.5243
Proposed	0.9011	0.8423	0.6982	**0.8177**	**0.7898**	**0.6922**	**0.5148**	0.3170	0.8695	**0.7215**	0.8663	**0.7300**

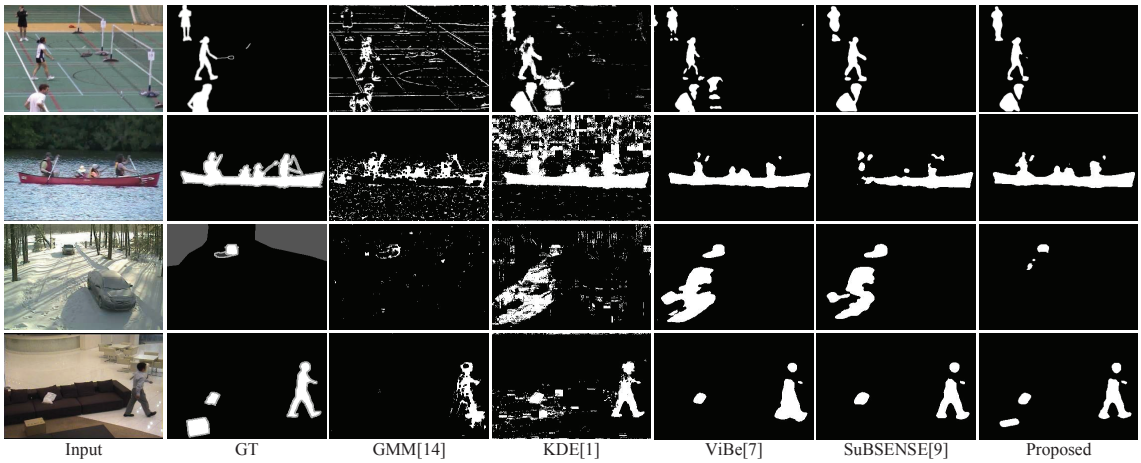

Figure 3: Visual comparison of foreground detection results.

ing the initial value of model parameters. What's more, we present a selective diffusion scheme to solve the problem brought by intermittent moving objects, which simply relies on distance analysis between foreground and surrounding background pixels. Experimental results show that our approach outperforms the state-of-the-art approaches in CDnet 2014.

5. ACKNOWLEDGMENT

This work was supported by 863 Program 2014AA015104, and National Natural Science Foundation of China 61273034, and 61332016.

6. REFERENCES

[1] E. Ahmed, H. David, and D. Larry. Non-parametric model for background subtraction. In *ECCV*, pages 751–767. Springer, 2000.

[2] Y. Chen, J. Wang, and H. Lu. Learning sharable models for robust background subtraction. In *ICME*. IEEE, 2015.

[3] Y. Chen, J. Wang, and H. Lu. Multiple features based shared models for background subtraction. In *ICIP*. IEEE, 2015.

[4] S. Chris and G. W. E. L. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2. IEEE, 1999.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *CVPRW*, pages 38–43. IEEE, 2012.

[6] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185, 2005.

[7] B. Olivier and V. D. Marc. Vibe: a powerful random technique to estimate the background in video sequences. In *ICASSP*, pages 945–948. IEEE, 2009.

[8] A. Sobral. BGSLibrary: An opencv c++ background subtraction library. In *IX Workshop de Visao Computacional*, Rio de Janeiro, Brazil, Jun 2013.

[9] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin. Flexible background subtraction with self-balanced local sensitivity. In *CVPRW*, pages 414–419. IEEE, 2014.

[10] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *Image Processing, IEEE Transactions on*, 24(1):359–373, 2015.

[11] M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012.

[12] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *CVPRW*, pages 387–394, 2014.

[13] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. In *CVPR*, pages 1–8. IEEE, 2007.

[14] Z. Zoran. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, volume 2, pages 28–31. IEEE, 2004.