BACKGROUND SUBTRACTION THROUGH MULTIPLE LIFE SPAN MODELING

Junliang Xing, Liwei Liu, Haizhou Ai

Computer Science and Technology Department, Tsinghua University, Beijing 100084, China
{xjl07,llw09}@mails.tsinghua.edu.cn, ahz@mail.tsinghua.edu.cn

ABSTRACT

Background subtraction plays a key role in many surveillance systems. A good background subtractor should not only be able to robustly detect targets under different situations (e.g. moving and static), but also to adaptively maintain the background model against various influences (e.g. dynamic scenes and noises). This paper proposes a novel background modeling approach with these good characteristics. By introducing the "life span" concept into a background model, different properties of the scene are obtained through different life span models. Specifically, three different models, i.e., the Long Life Span Model, the Middle Life Span Model, and the Short Life Span Model, are online adaptively built and updated in a collaborative manner. Output of the system gives an adaptive, robust, and efficient estimation of the foreground region which can facility many practical applications. Experiment results on lots of surveillance videos demonstrate the superiority of the proposed method over competing approaches.

Index Terms— Background subtraction, life span modeling, visual surveillance

1. INTRODUCTION

Object detection is of fundamental importance for many visual surveillance applications [1, 2]. Background subtraction provides an efficient way to perform this job. By subtracting background image from the input image and then thresholding the difference image, the moving objects could be identified. The effectiveness of a background subtraction method is heavily relied on the background model and the thresholds. For a short video sequence captured in a simple scene, a fixed background model and a fix global threshold can be sufficient to detect the moving objects. For real-world surveillance scenes, however, this choice is likely to fail since the background often changes gradually and sometimes even suddenly. More sophisticated techniques therefore are needed.

To compensate for the background changes, many previous algorithms use a constant rate to update the background model [1, 2]. In [2], distribution of each pixel is represented by a Mixture of Gaussians (MoG) and an algorithm is adopt to update the Gaussian component. Pixel which associates with uncommon Gaussian or matches no Gaussian is judged as foreground. The association and the match process usually depend on a given threshold. The approach has been applied in many systems and actually becomes a standard for background modeling [3]. The main problem, however, is that the potential inefficiency of the constant updating rate. What is more, this method could also wrongly update a foreground object into the background when it stays at a place for a while. To build a robust background model in dynamic scenes, methods like Bayesian decision [4], non-parametric model [5] and multi-feature subtraction [6] are proposed. Since the robustness of these methods often comes at the cost of the efficiency, they are not as widely used as [2] in practical systems.

In this paper, we present a new background modeling mechanism that is able to surmount most of the problems in existing methods and can be efficiently integrated into a visual surveillance system. Inspired by the work in [7], we introduce the life span concept into the background modeling process and collaboratively build multiple models with different life spans. Based on the distinct prosperities of a pixel deduced from different life span models, adaptive updating schemes are applied on its background models which makes them converge quickly. The proposed algorithm not only is able to successfully detect moving background components and static foreground objects, but also can work robustly under both gradual and sudden illumination changes.

2. THE PROPOSED APPROACH

The life span of a background model defines both its learning period and service period (Fig. 1). By building background models of different life spans, a lot of useful information about the scene and object can be obtained. What is more, by passing information between these background models, they can be built and updated more efficiently. Based on these observations, we propose to do background subtraction through multiple life span modeling. Fig. 1 gives an illustration of our life span modeling approach and the following parts will provide a more detailed description.

2.1. Model Description

Generally, a life span background model can be represented by its descriptor $\theta(t)$ which describes the background of the scene, its learning rate $\eta(t)$, and its threshold set $\tau(t)$ used for foreground segmentation. Therefore, we formalize a life span background model as:

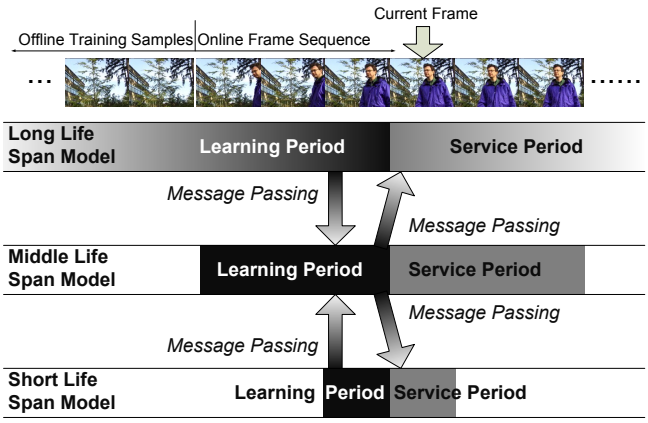

Fig. 1. Life span background modeling.

$$M = \{\theta(t), \eta(t), \tau(t)\}. \quad (1)$$

It should be noted that, although many previous methods often treat at least part of these three elements as constants, here we represent them all as a function of the time t since each of them can be made adaptive to different video frames. As shown in Fig. 1, we build three life span models for background subtraction: the Long Life Span Model (LLSM), the Middle Life Span Model (MLSM), and the Short Life Span Model (SLSM). Each of them captures different scene and object information based on its particular learning period and service period.

2.1.1. Long Life Span Model

If we observe a video sequence for a long time, we will find that pixels in every position have enough chances to obtain sufficient data generated by the background. Based on this observation, the LLSM uses a long period to learn a robust and complete background model which contains no object pixel and is able to captures multiple modes of the background scene. Corresponded to the learning time, the service time of LLSM is also very long. Denoting the LLSM as $M_L = \{\theta_L(t), \eta_L(t), \tau_L(t)\}$, its descriptor can be written as:

$$\theta_L(t) = \{I_{L,1}(t), I_{L,2}(t), \ldots, I_{L,K_L}(t)\}, \quad (2)$$

where $I_{L,k}(t) = \{x_{L,k,n}(t)\}_{n=1}^{N}$ is the background image with mode k at frame t, N is the total number of the image pixels. Usually $K_L = 2$ is sufficient for most of the scene which describes each background location with two different modes. The learning rate $\eta_L(t)$, for most of the time instance, can be set as a small constant value, e.g. $\eta_L(t) = 1/(10 \times 60 \times 30)$ which means the background model will be fully updated once in about 20 min. for a 30fps video sequence. But when a sudden change of the scene is detected, is can be temporal set with a large value to enable quick adaption to background modeling switching. The threshold set $\tau_L(t)$ here only contains one difference threshold $D_L(t)$ that is used to detect the foreground pixels in the difference image. It is decided by an adaptive scheme which will be described in Section 2.2.

2.1.2. Middle Life Span Model

The MLSM captures the scene and object information among the most recent frames which are often paid more attention by the high level system, e.g., object analysis, event prediction, etc. We use the MoG model as in [2] to describe the distribution of each pixel. Denoting the MLSM as $M_M = \{\theta_M(t), \eta_M(t), \tau_M(t)\}$, its descriptor can be represented as:

$$\theta_M(t) = \{\sum_{k=1}^{K_M} \omega_{n,k}(t) G_{n,k}(\mu_{n,k}(t), \Sigma_{n,k}(t))\}_{n=1}^{N}, \quad (3)$$

where K_M is the component number of the MoG model, $\omega_{n,k}(t)$, $\mu_{n,k}(t)$ and $\Sigma_{n,k}(t)$ are the weight, mean and covariance matrix of the kth Gaussian in the mixture (for clarity, the subscript n will be suppressed when there is no ambiguity). Unlike a constant updating rate used in [2], we employ an adaptive learning rate for $\eta_M(t)$ which can be written as:

$$\eta_M(t) = C_0 \beta(t) \alpha(t). \quad (4)$$

Here C_0 is the constant learning rate in [2]. $\beta(t)$ is a boot term which makes the learning rate faster at the system initialization stage:

$$\beta(t) = \begin{cases} 1/(C_0 t), & t < 1/C_0 \\ 1, & t \geq 1/C_0 \end{cases}. \quad (5)$$

$\alpha(t)$ is an adaption term which makes our MoG background modeling approach distinctly different from other methods. The details of its computing will be elaborated in Sec. 2.2.

For a standard MoG background modeling routing [2], it involves four thresholds, the Gaussian match threshold T_0, the background association threshold B_0, and the initial weight W_0 and variance V_0 for a new Gaussian. In our MLSM, these four thresholds are viewed as a function of time t in the threshold set which can be represented as:

$$\tau_M(t) = \{T(t), B(t), W(t), V(t)\}. \quad (6)$$

Determination of these adaptive threshold functions will be detailed in Section 2.2.

2.1.3. Short Life Span Model

The learning period and the service period of SLSM are both one frame which put it in a position to capture the motion changes between two consecutive frames. Here we employ an efficient frame differencing procedure with adaptive thresholding to detect the moving pixels between two consecutive frames. So the descriptor in SLSM represented as $\theta_S(t) = \{I_S(t)\}$ where $I_S(t)$ is the image data at frame $t-1$, the learning rate $\eta_S(t)$ constantly equals one, and the threshold set $\tau_S(t) = \{D_S(t)\}$ where the adaptive difference threshold function $D_S(t)$ is determined similarly to $D_L(t)$.

2.2. Model Building and Updating

We build and update the three life span models in a collaborative manner and all of them can be automatically learned online (although initializing the LLSM with a few offline samples could make it converge more quickly). Among the three life span models, the MLSM lies at an important position

Table 1. Adaptive building and updating of the MLSM.

LLSM	SLSM	MLSM	Operation ($d_k(t) = \|x(t) - \mu_k(t)\|$)
Object	Moving	MBG	$\alpha(t) = \exp(-d_k(t)/\sigma_k(t))$
		MFG	$\alpha(t) = \exp(-d_k(t)/\sigma_k(t))$
		MNG	$W(t) = 0.5W_0, V(t) = 2V_0$
	Static	MBG	$\alpha(t) = (1 + \exp(-d_k(t)/\sigma_k(t)))/2$
		MFG	$\alpha(t) = (1 + \exp(-d_k(t)/\sigma_k(t)))/2$
		MNG	$W(t) = W_0, V(t) = V_0$
Scene	Moving	MBG	$\alpha(t) = \exp(-d_k(t)/\sigma_k(t))$
		MFG	$\alpha(t) = (1 + \exp(-d_k(t)/\sigma_k(t)))/2$
		MNG	$W(t) = W_0, V(t) = V_0$
	Static	MBG	$\alpha(t) = (1 + \exp(-d_k(t)/\sigma_k(t)))/2$
		MFG	$\alpha(t) = (2 + \exp(-d_k(t)/\sigma_k(t)))/3$
		MNG	$W(t) = 2W_0, V(t) = 0.5V_0$

which both receives and sends messages from and to other two models (Fig. 1) and whose subtraction result is used as the final result of the system.

The main problem of a standard MoG background modeling procedure is the inefficiency of its model learning which often updates a static foreground object mistakenly into the background and leave a "ghost" in the place where an object has just left. In our collaborative life span modeling process, the LLSM can provide the information whether a pixel is an object pixel or scene pixel while the SLSM can provide the information whether a pixel is moving or static. With these useful and explicit messages, we can dynamically initialize new Gaussian models and adaptively update existing Gaussian models. This process involves the determination of the adaption term of $\alpha(t)$, the initial weight $W(t)$ and variance $V(t)$ for a new Gaussian model (since the Gaussian match threshold $T(t)$ and background association threshold $B(t)$ are relatively stable in most of the scene, we use their suggested values as in [2, 3]). In Table 1, we summarize the adaptation rule under different situations. Here, MBG means the input pixel matches a background Gaussian, MFG means it matches a foreground Gaussian, and MNG means it matches no Gaussian. From Table 1 we can see that, our MLSM modeling process does not introduce any new thresholds and meanwhile greatly improves the adaptability of the standard MoG method since a static scene pixel gets much larger initialization and updating rate than a moving object pixel.

The LLSM is initialized using the first frame or a few offline training samples and then gradually updated online based on the responses of MLSM. Specifically, if a pixel's response in MLSM is MBG, the corresponding background image pixel in $\theta_L(t)$ is updated:

$$x_{L,l}(t+1) = (1 - \eta_L(t))x_{L,l}(t) + \eta_L(t)x(t), \quad (7)$$

where l is the index of the matched background image in $\theta_L(t)$ which is calculated as:

$$l = \underset{k}{\operatorname{argmin}}\{|x(t) - x_{L,k}(t)|\}_{k=1}^{K_L}. \quad (8)$$

This updating strategy only uses the background pixels that have been well validated by the MLSM to update the LLSM. The resulting LLSM is therefore robust to noises and capable

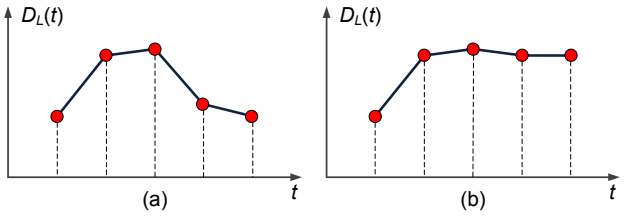

Fig. 2. Different modes of illumination changes: (a) "once off" change; (b) "switching" change.

of capturing long time scene information. For the segmentation threshold $D_L(t)$, it is also adaptively updated by the responses of the MLSM. Supposing the number of background pixels detected by MLSM is $N_M(t)$, we first calculate the *cumulative histogram* of the difference image in the LLSM which can be represented as $H(t) = \{h_i(t)\}_{i=1}^{N_B}$($N_B$ is the bin number which can be set as 256 for grey image and 765 for color image). $D_L(t)$ is decided by following equations:

$$D_L(t) = \begin{cases} D, & h_{D+1}(t) - N_M(t) \geq N_M(t) - h_D(t) \\ D+1, & \text{otherwise} \end{cases}, \quad (9)$$

where D satisfies the following inequalities:

$$\begin{cases} h_{D+1}(t) \geq N_M(t) \\ h_D(t) \leq N_M(t) \end{cases}. \quad (10)$$

The basic assumption here is that the number of background pixels may not change greatly in consecutive frames. Since this is common for real-world video sequences, the threshold function $D_L(t)$ therefor can adapt to different videos.

For the SLSM, the building and updating process is much easier. At every frame, it updates its descriptor using previous frame data and set the segmentation threshold $D_S(t)$ function similarly to $D_L(t)$.

2.3. Illumination Change Detection

Another capability of the threshold function $D_L(t)$ is to detect global illumination changes. If their values changes greatly (typically > 10) in two consecutive frames, usually the global illumination state is changing. In this situation, we propagate the subtraction result of previous frame to current frame and detect the two modes showed in Fig. 2 to classify "once off" change an "switching" change by checking the values of the threshold functions in five frames. If the change is confirmed to be a "switching" change, a new background image will be added to descriptor in LLSM.

3. EXPERIMENTS

The proposed method has been implemented in C++ and evaluated on many different sequences collected from publicly available datasets like PETS 2009 [8], Wallflower [9] and CAVIAR [10]. We compare our method with the state-of-the-art MoG method [2] and Bayesian decision method [4] implemented in [3]. All the experiments are carried on a PC with an Intel Core Quad 2.40 GHz CPU and 4G RAM.

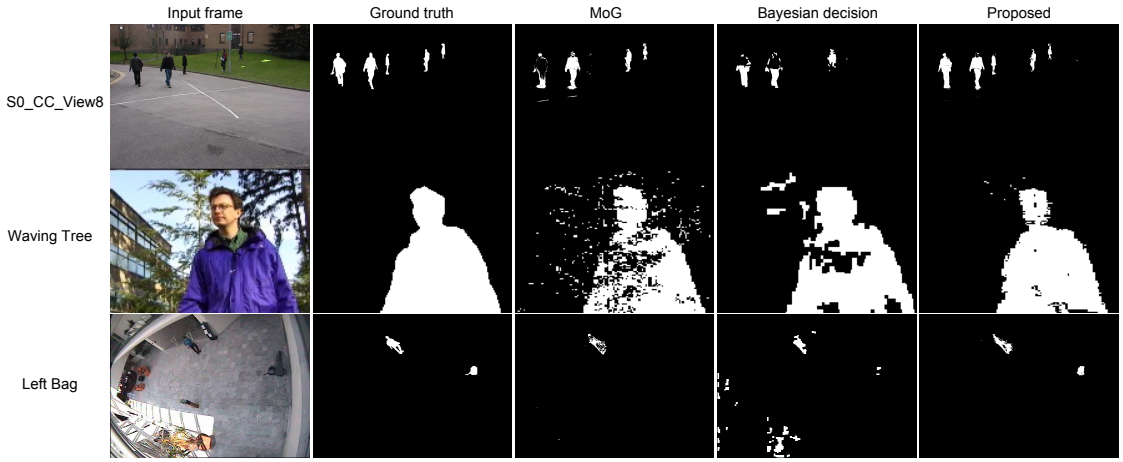

Fig. 3. Background subtraction results on three different video sequences.

Table 2. Speed Comparison.

Algorithm	Video resolution		
	160 × 120	320 × 240	640 × 480
MoG [2]	96fps	20fps	9fps
Bayesian decision [4]	35fps	13fps	4fps
Proposed	180fps	32fps	12fps

In Fig. 3, we give the qualitative result of the three algorithms with default parameters on three typical sequences from the datasets. In the *S0_CC_View8* sequence, our method has detected the middle person while other two methods fail to detect him since he has stayed there for a long time. In the *Waving Tree* sequence, our method adapts to the dynamic background much better than other two methods. Our method also successfully detected the left bag which is fully missed by the MoG method in the *Left Bag* sequence.

To evaluate our method quantitatively, we manually label the foreground mask of the *S0_CC_View8* sequence every 30 frames to get 25 ground-truth images. We generate different subtraction results of the three algorithms with different thresholds on these images and compute their precision and recall. In Fig. 4, the ROC curves of these methods are plotted to give a quantitative comparison. It can be easily observed that our method significantly outperforms other two methods.

We further evaluate the efficiency of the three algorithms by running them on videos with different resolutions. Table 2 summarizes the running times. Our algorithm runs even faster than the MoG method implemented by [3] and could be used for real-time applications after further optimization.

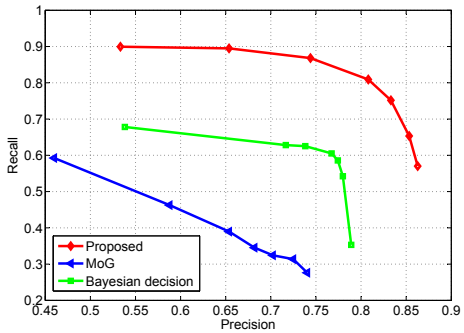

Fig. 4. Quantitative evaluation of the three methods.

4. CONCLUSION

In this paper, a novel background subtraction technique is presented in which the "life span" concept is used in the background modeling. By building life span models in a collaborative manner, the proposed approach can adaptively, robustly, and efficiently detect foreground objects in different video scenes. Experiment results demonstrate its significant improvements over the state-of-the-art methods.

5. ACKNOWLEDGEMENT

This work is supported by National Science Foundation of China under grant No.61075026, and it is also supported by a grant from Omron Corporation.

6. REFERENCES

[1] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, 2006.

[2] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, 2000.

[3] "OpenCV," http://www.sourceforge.net/projects/opencvlibrary.

[4] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *ACM Int. Conf. Multimedia*, 2003.

[5] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Eur. Conf. Comput. Vision*, 2000.

[6] M. Azab, H. Shedeed, and A. Hussein, "A new technique for background modeling and subtraction for motion detection in real-time videos," in *IEEE Int. Conf. Image Processing*, 2010.

[7] Y. Li, H. Ai, T.Yamashita, S.Lao, and M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1728–1740, 2008.

[8] "PETS 2009 dataset," http://www.cvg.rdg.ac.uk/PETS2009.

[9] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *IEEE Int. Conf. Comput. Vision*, 1999.

[10] "CAVIAR dataset," http://homepages.inf.ed.ac.uk/rbf/CAVIAR.