

# Background Subtraction: Model-Sharing Strategy Based on Temporal Variation Analysis

Yufeng Chen, Kun Zhao<sup>(✉)</sup>, Wenzhe Wu, and Shikai Liu

Beijing Institute of Technology, Zhongguancun Road, Haidian District,  
Beijing, People's Republic of China  
zhkflame@163.com

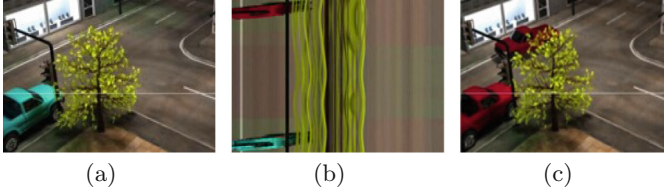
**Abstract.** This paper presents a new approach for moving detection in complex scenes. Different with previous methods which compare a pixel with its own model and make the model more complex, we take an iterative model-sharing strategy as the process of foreground decision. The current pixel is not only compared with its own model, but may also compared with other pixel's model which has similar temporal variation. Experiments show that the proposed approach leads to a lower false positive rate and higher precision. It has a better performance when compared with traditional approach.

## 1 Introduction

For many computer vision applications, detecting moving objects from a video sequence is a critical and fundamental part. A common approach to detect moving objects is background subtraction. Background subtraction have been widely used for video surveillance, traffic management, tracking and many other applications.

The purpose of a background subtraction algorithm is to distinguish moving objects from the background. A good background model should be able to achieve some desirable properties: accurate in shape detection, reliable in different light conditions, flexible to different scenarios, robust to different models of background, efficient in computation [1]. In developing a good background subtraction algorithm, there are many challenges such as dynamic scenes, illumination changes, shadows and other limitations. Researchers have been devoted to develop new methods and techniques to overcome different limitations. Different methods have been proposed to solve difference issues and challenges.

In order to deal with complex scenes, most traditional methods take the strategy of making each pixel's model contain more information. This strategy may lead to the model of each pixel too complex, and still not contain enough information because of the random movement. These methods all compare the current pixel with its own models. We take another strategy that the current pixel may compared with other pixels model which has similar temporal variation. In order to determine whether they are similar to each other, we use stability which is get by wavelet transform to quantify the pixel's variation over



**Fig. 1.** Temporal variation of a chosen line

time. Figure 1 shows the temporal variation of a chosen line which is marked as white. Figure 1(a) is the frame at time  $a$  and Fig. 1(b) is the frame at time  $b$ . Figure 1(c) represents the variation of the chosen line from time  $a$  to  $b$ . We can see different variations in different areas. The dynamic background such as waving trees changed more obviously and this means their variation is less stable.

In this paper, we present a new approach which use model-sharing strategy based on temporal variation analysis. In Sect. 2, we review the previous work related to background subtraction. Section 3 describes our methods in detail. Section 4 discusses experimental results including comparison with other algorithms.

## 2 Related Works

Numerous methods for detecting moving objects have been proposed in the past. The most direct method to identify moving objects is using frame differences which focus on the changes between two frames. Large changes are considered as foreground.  $W^4$  [2] model use the variance found in a set of background images with the maximum and minimum intensity value and the maximum difference between consecutive frames. A very popular method to model each pixel is to use the Gaussian distribution. Pfinder [3] assumes that at a particular image location the pixels over a time window are single Gaussian distributed. Mixture of Gaussians(MoG) [4] is a widely used approach for background modeling. Here each pixel is modeled as a mixture of weighted Gaussian distributions. Pixels which are detected as background are used to update the Gaussian mixtures by an update rule. These methods can deal with small or gradual changes in the background, but if the background includes only one static scene, they may fail when background pixels are multi-modal distributed or widely dispersed in intensity [1]. Such methods lack the adaptability to the dynamic scenes. Lee [5], Tuzel [6] and other researchers have improved MOG in different aspects. More detailed reviews can be find in [7]. Kim [8] propose an approach for detecting moving objects by building an codebook model. The method can deal with periodic variations over time, but it is not able to capture complex distributions of background values.

Over the past years, the non-parametric method has been extensively studied. A non-parametric method was proposed by Elgammal [9] to estimate the

density function for each pixel from many samples by making use of kernel density estimation technique. Wang [1] propose SACON, in which the model of each pixel include a history of the  $N$  most recent pixel values and the background model values are updated using a First-in First-out strategy. Barnich and Droogenbroeck [10] propose ViBe and this approach learn and update the back ground model of each pixel based on a random strategy. The PBAS [11] use a history of  $N$  image value as the background model as in SACON and use a similar random update rule as in ViBe. They all belong to the non-parametric method. Our method also follow a non-parametric paradigm, further more, we add the model-sharing strategy to detect moving objects iteratively.

### 3 Model-Sharing Strategy

#### 3.1 Regional and Temporal Analysis

In [12], “case-by-case model sharing” is used for bidirectional analysis. A given background model is selected from a database and it is not always selected for the same pixel, moreover, it could be shared by several pixels. We also use the model sharing method, but we didn’t use a database to query a suitable model. Our model-sharing strategy is based on two concepts.

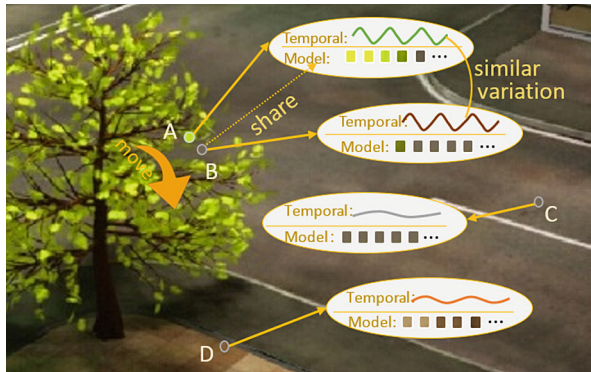


Fig. 2. Example of using model-sharing strategy

1. Different background areas have different stability, same region has similar stability. The stability is used to describe temporal variation. For instance, stable background such as buildings and roads have no change or change a little, so they have a higher stability. Dynamic background such as waving trees changed dramatically over time due to the winds, so they have a lower stability, and this lead to a higher false positives rate.

2. The movement of dynamic background is random and regional. Such as grass, trees, and rivers, they move randomly in a region. Due to the randomness, the model of a pixel in this region may not contain enough information and this could affect the accuracy of segmentation.

The two concepts will help us to find the models that could be shared. Figure 2 shows an example of using model-sharing strategy. The pixels A and B in the dynamic area, the leaves may move to their positions. In model learning process, the leaf at A, the model of A could contain the leaf’s color information. If the leaf moved from A to B, because B’s model didn’t contain the leaf’s information and this may lead to error detection. A and B are both trajectory of the leaf, if they can share their models so that the current pixel B could share the model of A, it may reduce error detection. But we don’t know which model should be shared. In order to simulate the random motion, we use the reverse speculation. Randomly and iteratively chose a pixel around B, the chosen pixel’s model may include the leaf color information or not. If they have a similar temporal variation, compare B with the chosen pixel’s model. If they match successfully, B should be detected as background, otherwise chose another pixel which has a similar variation to match B again until reaching the iterative boundary.

### 3.2 Quantify the Variation Over Time

We mentioned that different background area has different variations over time. In order to distinguish different areas for model-sharing, we need to quantify the variation in time-domain by stability.

$x^t$  represent a pixel  $x$  at time  $t$ . First, we judge if  $x^t$  is stable by using  $B(x)$ .  $B(x)$  is preserved by an array of  $k$  recently observed pixel values which was detected as background.

$$B(x) = \{B_1(x), B_2(x), \dots, B_k(x) | B_i(x) \in background, 1 \leq i \leq k\} \quad (1)$$

Then, we can get different frequency coefficients  $\{C_1, C_2, \dots, C_m\}$  by taking Haar wavelet transform on  $B(x)$ . The coefficients could represent how much the pixel  $x^t$  change in different frequency. We assume that if all the coefficients are in a given range,  $|C_i| < R_{coe}$ ,  $1 < i < m$ , then  $x^t$  should be judged as stable, otherwise it should be judged as unstable.

$$s(x^t) = \begin{cases} 1, & \text{if } |C_1| < R_{coe} \cap |C_2| < R_{coe} \cap \dots \cap |C_m| < R_{coe} \\ -1, & \text{otherwise} \end{cases} \quad (2)$$

We use  $S_x^t$  to describe the stability of the pixel  $x$  at time  $t$ .

$$S_x^t = \sum_{i=0}^n s(x^i) \quad (3)$$

$S_x^t$  represent the accumulation of  $s(x^i)$ . We define the upper and lower bound for  $S_x^t$ ,  $0 < S_x^t < R_{max}$ .  $R_{max}$  represent the maximum of  $S_x^t$ , so that  $S_x^t$  is in a reasonable bounds. The larger the  $S_x^t$  is, the pixel  $x$  is more likely in stable areas. Figure 3 shows the distribution of  $S_x^t$  in different scenes. Dynamic background could be distinguished effectively.



**Fig. 3.** Distribution of  $S_x^t$  in different scenes

### 3.3 Segmentation Decision

The use of model-sharing strategy will simplified the background model. In our method, the background model  $M(x)$  s only initialized by the first  $n$  frames.

$$M(x) = \{M_1(x), M_2(x), \dots, M_n(x)\} \quad (4)$$

Different from other methods, our segmentation decision is iterative for model-sharing. In the current frame, not every pixel need to share other pixels models, only the pixel detects as foreground should be concerned share others. Because if a pixel is detected as background means that its model has enough information to match the current value, so it need not to be detected again. A lower  $S_x^t$  means  $x$  is in a more dynamic area which has a higher false positives rate, so it needs more information. The number of iteration  $L_x$  is decided by  $S_x^t$ . We define  $0 < L_x < R_{iter}$ . Here,  $L_x = 1$  imply that no other models is shared,  $R_{iter}$  represent the maximum iterations.

$$L_x = \lfloor \frac{R_{max}}{S_x^t} \rfloor \quad (5)$$

$F(x, M(x'))$  represent the pixel  $x$  is compared with the model of  $x'$ ,  $x'$  is a random chosen pixel around  $x$  satisfied  $|S_x^t - S_{x'}^t| < R_{stab}$ ,  $R_{stab}$  is a given threshold.  $F(x, M(x))$  represent compare  $x$  with its own model.

If the distance between the current pixel value  $I(x)$  and  $M_k(x')$  is smaller than a decision threshold  $R_d$ , we decide they can match each other.

$$F(x, M(x)) = \begin{cases} Background, & \text{if } \#(x, M(x')) < \#_{min} \\ F(x, M(x')), & \text{else if } L_x > 1 \\ Foreground, & \text{else} \end{cases} \quad (6)$$

We use  $\#(x, M(x'))$  represent the match number when  $x$  is compared with  $M(x')$ . If  $\#(x, M(x'))$  ia larger than a given threshold  $\#_{min}$  the pixel  $x$  should be detected as background. Otherwise, if  $L_x > 1$ ,  $x$  should be detected iteratively to share other pixels model until  $L_x$  reach the boundary value.

### 3.4 Update Mechanism

The update mechanism is essential to adapt to changes over time, such as lighting changes, shadows and moving objects.

We use a similar random update mechanism as in ViBe. For a pixel  $x$ ,  $M_i(x)(i \in 1 \cdots N)$  is randomly chosen from its model  $M(x)$ . The value of  $M_i(x)$ , is replaced by the current pixel value in a random time subsampling. This mechanism allows the current value to be learned into the background model and ensures a smooth decaying lifespan for the samples stored in the background pixel models. The current value could also update the model which has be shared with  $x$  successfully. For example, when  $x$  is compared with the model of  $x'$  and detected as background, the current value could also update  $M(x')$ . The propagate rules is different from ViBe's that randomly chosen a neighborhood pixel to update.

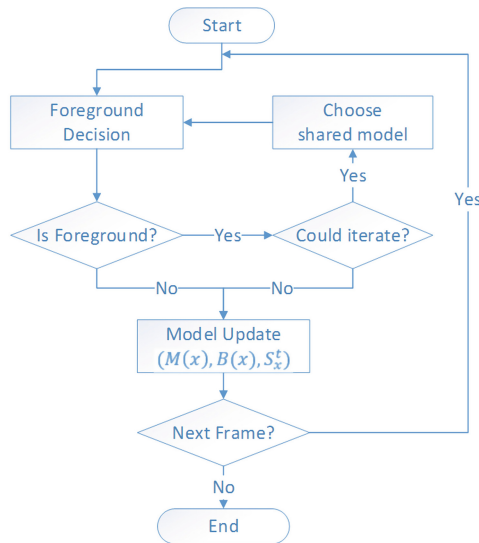


Fig. 4. The process flow of the proposed method

Figure 4 shows the process flow of the proposed method. The foreground decision is iterative until  $L_x$  reach the boundary value. If it meets the iterative condition, a random chosen model will be shared for foreground decision.  $B(x)$  is updated to calculate  $S_x^t$ ,  $S_x^t$  is used to choose a suitable shared model.  $M(x)$  is updated to adapt to environmental changes.

## 4 Experimental Results

### 4.1 Determination of Parameters

As mentioned in previous discussions, there are some parameters in our approach. Since the approach is evaluated on the open database, we seek an optimal set of parameters that give the best balanced performance in different categories.

$K = 8$ :  $K$  is the number of recent observed pixel values which was detected as background.  $K$  need to be the integer power of 2 for taking Haar wavelet transform. In order to reflect the stability of a pixel and reduce the computational complexity,  $K = 8$  is suitable.

$N = 24$ :  $N$  is the number of samples in background model. The higher values of  $N$  provide a better performance. In the interval between 20 and 30, it will lead to best performance. They tend to saturate for values higher than that interval.

$\#_{min}$ : The number of matched samples need to be greater than or equal to  $\#_{min}$  when the current pixel is detected as background. An optimum is found at  $\#_{min} = 2$ . The same optimal value has been found in [10].

$R_d = 20$ :  $R_d$  used as a threshold to compare a new pixel value with pixel samples. The same optimal value also has been found in [10].

$R_{coe} = 5$ :  $R_{coe}$  is used to measure the different frequency coefficients to judge if the current pixel is stable at time  $t$ . The larger the  $R_{coe}$  is, the more likely the current pixel judged as stable.

$R_{max} = 250$ :  $R_{max}$  represent the maximum of  $S_x^t$  so that  $S_x^t$  in a reasonable bounds.

$R_{iter} = 7$ :  $R_{iter}$  represent the maximum iteration, The larger the  $R_{iter}$  is, the more models may be shared and need more calculations.

$R_{stab} = 30$ :  $R_{stab}$  is used to measure if the pixel  $x$  and a randomly chosen pixel  $x'$  have similar temporal variation, so as to judge if  $x$  could share the model of  $x'$  or not.

### 4.2 Performance Evaluation

The results are evaluated by some metrics including FPR, FNR, Recall, Precision and F-measure.

$$FPR = \frac{FP}{FP + TN} \quad FNR = \frac{FN}{TP + FN} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad Precision = \frac{TP}{TP + FP} \quad (8)$$

$$F - measure = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (9)$$

The proposed method is compared with some typical methods provided in BGSLibrary [13] on the SABS dataset [14] which includes a multitude of challenges for general performance overview such as illumination changes, dynamic background and shadows.

**Table 1.** Performance compared with other approaches

	FPR	FNR	Recall	Precision	F-measure
MOG [15]	0.014	0.251	0.748	0.573	0.649
GMG [16]	0.047	0.117	0.882	0.323	0.473
MultiLayer [17]	0.014	0.177	0.822	0.595	0.691
ViBe [10]	0.013	0.251	0.749	0.596	0.664
PBAS [11]	0.013	<b>0.091</b>	<b>0.897</b>	0.617	0.735
Ours	<b>0.006</b>	0.276	0.753	<b>0.751</b>	<b>0.752</b>

Table 1 shows the results of different metrics when compared with other methods. It can be seen that our method has the best performance in Precision, FPR and F-measure. In other performance metrics, our methods has an average performance.

We also evaluate our approach on eleven real scenes from different categories in CDNET video dataset [18]. This dataset include scenarios with baseline, dynamic background, camera jitter intermittent object motion, shadow thermal, bad weather, low frame-rate, night videos, PTZ and turbulence.







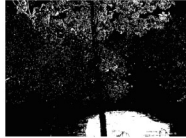










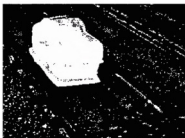


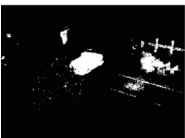
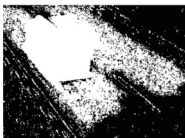


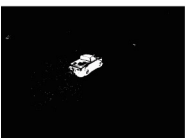
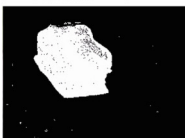


**Table 2.** Performance in different scenarios

	FPR	FNR	Recall	Precision	F-measure
Baseline	0.0002	0.087	0.912	0.974	0.942
Dynamic background	0.0009	0.412	0.581	0.910	0.714
Camera jitter	0.0005	0.383	0.616	0.984	0.758
Intermittent object motion	0.0021	0.551	0.449	0.893	0.598
Shadow	0.0002	0.216	0.783	0.894	0.876
Thermal	0.0051	0.328	0.671	0.926	0.778
Bad weather	0.0001	0.283	0.716	0.998	0.834
Low framerate	0.0011	0.488	0.511	0.887	0.649
Night videos	0.0092	0.289	0.710	0.685	0.697
PTZ	0.0021	0.381	0.612	0.814	0.704
Turbulence	0.0001	0.236	0.763	0.778	0.771
Over all	0.0019	0.332	0.667	0.895	0.756

Table 2 results performance of the proposed approach in all eleven scenarios. Our method has best performance on Precision and FPR for all categories. This is because the process of foreground decision is iterative in model-sharing strategy and it can improve accuracy. Table 3 shows comparative segmentation maps in different scenarios by using different methods.



**Table 3.** Comparative segmentation maps in different scenarios

	basic	jitter	dynamic	thermal
Input				
MOG				
GMG				
Multi				
ViBe				
PBAS				
Ours				

## 5 Conclusion

In this paper, a model-sharing strategy based on temporal variation analysis is proposed for moving detection in complex scenes. The current pixel is not only compared with its own model, but may also be compared with the model of other pixel which has similar temporal variation. This method could offset the lack of information for a pixel. Comparison with previous methods shows that the proposed approach has better performance on precision and FPR, this

lead to better performance on F-measure which seem to be best for a balanced comparison.

## References

1. Hanzi, W., David, S.: A consensus-based method for tracking: modelling background scenario and foreground appearance. *Pattern Recogn.* **40**, 1091–1105 (2007)
2. Haritaoglu, I., Harwood, D., Davis, L.S.: W 4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 809–830 (2000)
3. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfnder: real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 780–785 (1997)
4. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1999*, vol. 2. IEEE (1999)
5. Lee, D.S.: Effective gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 827–832 (2005)
6. Tuzel, O., Porikli, F., Meer, P.: A bayesian approach to background modeling. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, CVPR Workshops 2005*, p. 58. IEEE (2005)
7. Cristani, M., Farenzena, M., Bloisi, D., Murino, V.: Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP J. Adv. Sig. Process.* **2010**, 43 (2010)
8. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* **11**, 172–185 (2005)
9. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: *Vernon, D. (ed.) ECCV 2000. LNCS*, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
10. Olivier, B., Marc, V.D.: Vibe: a universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.* **20**, 1709–1724 (2011)
11. Hofmann, M., Tiefenbacher, P., Rigoll, G.: Background segmentation with feedback: the pixel-based adaptive segmenter. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 38–43. IEEE (2012)
12. Shimada, A., Nagahara, H., Taniguchi, R.I.: Background modeling based on bidirectional analysis. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1979–1986. IEEE (2013)
13. Sobral, A.: Bgslibrary: an opencv c++ background subtraction library. In: *IX Workshop de Viso Computacional (WVC 2013)* (2013)
14. Brutzer, S., Hoferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1937–1944. IEEE (2011)
15. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, vol. 2, pp. 28–31. IEEE (2004)
16. Godbehere, A.B., Matsukawa, A., Goldberg, K.: Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In: *American Control Conference (ACC)*, pp. 4305–4312. IEEE (2012)

17. Yao, J., Odobez, J.M.: Multi-layer background subtraction based on color and texture. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
18. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: a new change detection benchmark dataset. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8. IEEE (2012)