Motion Compensation Based Fast Moving Object Detection in Dynamic Background

Wei Zhang, Chenglong Li, Aihua Zheng(✉), Jin Tang, and Bin Luo

Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, AnHui University, Hefei 230601, China
{wzhang1127,lcl1314}@foxmail.com, {ahzheng214,tj}@ahu.edu.cn, luobinahu@163.com

Abstract. This paper investigates robust and fast moving object detection in dynamic background. A motion compensation based approach is proposed to maintain an online background model, then the moving objects are detected in a fast fashion. Specifically, the pixel-level background model is built for each pixel, and is represented by a set of pixel values drawn from its location and neighborhoods. Given the background models of previous frame, the edge-preserving optical flow algorithm is employed to estimate the motion of each pixel, followed by propagating their background models to the current frame. Each pixel can be classified as foreground or background pixel according to the compensated background model. Moreover, the compensated background model is updated online by a fast random algorithm to adapt the variation of background. Extensive experiments on collected challenging videos suggest that our method outperforms other state-of-the-art methods, and achieves 8 fps in efficiency.

Keywords: Fast object detection · Random algorithm · Dynamic background · Motion compensation

1 Introduction

Moving object detection with dynamic background is to detect moving objects under a moving camera, and has a broad prospect of application and research value in the intelligent transportation, medical diagnosis, security monitoring, and many other industries. However, due to the high complexity of the existing method which are unable to meet the time demand of many applications, it is still a challenging subject in computer vision.

Aimed at overcoming this limitation, this paper proposes a fast moving object detection framework in dynamic background, in which the motion compensation algorithm is utilized to accommodate the dynamic background, and the background model is updated online in a probability way to adapt the variation of background. Specifically, the background model of each pixel consists of a set of pixels, which are initialized by its location and neighbors. When new frame arriving, the optical flow algorithm, based on edge-preserving patch matching

H. Zha et al. (Eds.): CCCV 2015, Part II, CCIS 547, pp. 247–256, 2015.
DOI: 10.1007/978-3-662-48570-5_24

is employed to compensate the motion of each pixel and propagate their background models from previous frame to current one. Then, every pixel can be classified as the foreground or background pixel by the matching score with their background models. Furthermore, the background models are updated in an online fashion to adapt the variation of background.

To the best of our knowledge, it's the first time to develop a near real-time moving object detection in dynamic background. The key contributions of this paper are summed up in three aspects. Firstly, a general framework is proposed for robustly and fast detecting moving objects in dynamic background, in which the detection speed can reach near real-time. Secondly, a robust background model based on motion compensation is developed and updated online by a random algorithm to adapt the motion and variation of background over time. Thirdly, 10 challenging videos are collected in dynamic background from different scenes to comprehensively evaluate our approach against other state-of-the-art approaches. Extensive experiments on the collected challenging video sequences suggest that our method outperforms other state-of-the-art methods in accuracy, and achieves 8 fps in efficiency.

2 Related Works

Generally, moving object detection methods can be divided into two categories, *i.e.*, static background and dynamic background. At present, moving object detection in static background has become an increasingly mature technique and many related technologies have been successfully applied to real life. Stauffer et al. [11] proposed an adaptive background mixture models for real-time tracking, in which each pixel was modelled as mixture of Gaussian while using an online approximation to update the model. Some improved approaches on Gaussian Mixture Model (GMM) had proposed to address different issues, such as parameters initialization [7], model updating [8] and the number of Gaussian components [18]. Although these approaches achieved nearly real-time, it was still difficult to apply them to many applications unless with some parallel optimizations. Barnich et al. [5],[14] presented a simple background modelling method to detect the moving object with high accuracy and efficiency. The background model of each pixel consisted of a set of values taken in the past at the same location or in the neighborhood and randomly updated from the last pixel at same location or its neighbors. Although lots of progress has been made on moving objects detection in static background, there still exists many critical issues in dynamic background. Zhou et al. [17] proposed a moving object detection framework DECOLOR to address several complex scenarios, such as non-rigid motion and dynamic background. They assumed that the transformation between consecutive frames was linear and thus utilized the 2D parametric transforms [12] to model translation, rotation, and planar deformation of the background. DECOLOR can achieve state-of-the-art performance, but it was time-consuming and only processed the video in a batch fashion. Therefore, we aim at finding a kind of better way to solve some mentioned problems in dynamic background.

3 Our Approach

The details of our approach are described in this section. We utilize motion estimation algorithm to adaptively maintain a robust background models in the dynamic background. Fig. 1 shows the flowchart of our framework.

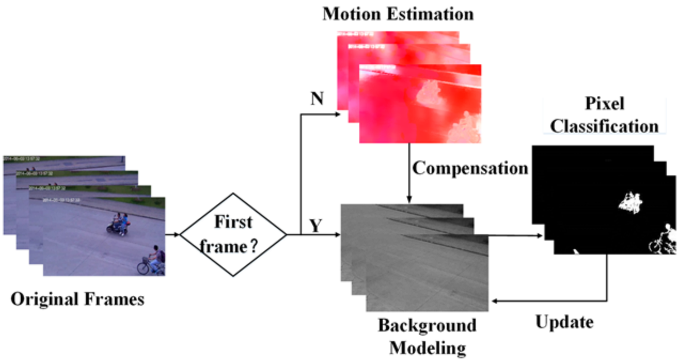

Fig. 1. Flowchart of our framework.

3.1 Motion Estimation

In this paper, the motion of each pixel will be accurately estimated to propagate to their background model to accommodate the motion of the camera. Most of existing methods on dense optical flow are time-consuming and computationally inefficient [1]. On the other hand, a fast optical flow algorithm based on edge-preserving PatchMatch is recently proposed by Bao et al. [3] with high accuracy and efficiency. Therefore, we employ the edge-preserving PatchMatch optical flow to estimate the motion of background in this work, and briefly review it as follows.

The edge-preserving PatchMatch optical flow is a fast algorithm that employs approximate the nearest neighbor field [6] to handle the large displacement motions and consists of four steps: matching cost computation, correspondence approximation, occlusions and outliers handling, and subpixel refinement.

(1) Matching Cost Computation. The edge-preserving PatchMatch optical flow follows the traditional local correspondence searching framework [10]. To make the nearest neighbor field preserve the details of the frame, it employs bilateral weights [16] into matching cost calculation, and can be defined as

$$d(a,b) = \frac{1}{W}\sum_{\Delta}\omega(a,b,\Delta)C(a,b,\Delta), \tag{1}$$

where a and b denote two pixels, Δ indicates the patches center on a and b , W is a normalization factor, $\omega(\bullet)$ is the bilateral weighting function and $c(\bullet)$ is the robust cost between a and n. More detailed definitions please refer to [3].

(2) Correspondence Approximation. To produce high-quality flow fields, this optical flow method utilizes self-similarity propagation and a hierarchical matching scheme to approximate the exact edge-preserving Patch Match[4]. Firstly, self-similarity propagation algorithm is based on the fact that adjacent pixels tend to be similar to each other. Specifically, for each pixel, a set of pixels from its surrounding region is randomly selected and stored into a self-similarity vector in the order of their similarities to the center pixel. Then, its adjacent pixels' vector is merged into its own vector from top-left to bottom-right. This process is reversely repeated. Thanks to the propagation between adjacent pixels, the algorithm can produce reasonably good approximate results in a much faster speed. Secondly, a hierarchical matching scheme is employed to further accelerate the algorithm and similar with SimpleFlow method [13].

(3) Occlusions and Outliers Handling. The edge-preserving PatchMatch optical flow explicitly performs the forward-backward consistency check [9] between the two nearest neighbor fields to detect occlusion regions. Moreover, a weighted median filtering is performed [2] on the flow fields to remove the outliers.

(4) Subpixel Refinement. The edge-preserving PatchMatch optical flow produces subpixel accurately with a more efficient technique - paraboloid fitting, which is a 2D extension from the 1D parabola fitting [15].

3.2 Background Modeling

Compared with the background models of a static background, the background modeling in dynamic background is difficult to maintain online since the background pixels are also moving. Although estimated optical flow can compensate the background motion, the background model is still sensitive to noises, due to incorrect optical flow estimation. Thus, a robust pixel model of background is proposed in this paper to adaptively detect the objects in the dynamic background. The two main components of the proposed background model can be described as follows.

Initialization. For each input video, the first frame is selected to initialize the background model. The background model of each pixel is a set of pixel values, and can be represented as

$$B(p) = \{I(p_1), I(p_2), \cdots, I(p_n)\}, \tag{2}$$

where $p_i \in N(p)$, and $N(\bullet)$ indicates the neighbors of pixel p. $I(\bullet)$ denotes the pixel value. For each pixel, n samples are selected from itself and its neighboring pixel values to initialize its background model.

Update. In this section, we assume that each pixel has been accurately classified by the background model (the details are discussed in next section) when new frame arriving. Thus, the background model of each pixel can be updated online by randomly selecting the classified background pixels at the same location or its neighbors. Specifically, for one classified background pixel p_b, two robust background model updating strategies are adapted to obtain its background model $B(p_b)$.

Firstly, one element from $B(p_b)$ is selected in a uniform probability way to replace p_b. Secondly, one pixel value is heuristically taken from its neighbors $N(p_b)$, and substituted by the element randomly selected in $B(p_b)$. Herein, we assume that if one pixel belongs to its background model, its distance to all the values of the background model should be as close as possible. This assumption will be helpful to suppress the effect of the noises. Thus, the selected probability of pixel p_b^i from $N(p_b)$ is defined as

$$q_i = \frac{1}{Q} \exp\{-\frac{1}{n} \sum_{j=1}^{n} D(I(p_b^i), B_j(p_b))\}, \tag{3}$$

where $D(\bullet, \bullet)$ denotes the Euclidian distance function, and Q is a normalization factor.

In addition, to accommodate the change speed of the background, the updating probability, called as updating factor and denoted as η in this paper, is introduced to determine whether the above updating is carried out or not.

3.3 Pixel Classification

Given the background model of previous frame, it can be propagated to the current frame by employing the motion estimation algorithm. Then, every pixel of current frame can be classified as the foreground or background pixel according to the matching scores with their corresponding background model.

For one pixel p, the matching score with background $B(p)$ is defined as

$$M(p) = \sum_{i=1}^{n} \delta(D(I(p), B_i(p)) > R), \tag{4}$$

where $\delta(\bullet)$ denotes the indicator function, and R indicates the adaptive threshold of matching cost, which is determined by the variation σ of $B(p)$. Herein, σ indicates the complexity of the background, and, R is defined as

$$R = \begin{cases} 20, \sigma/2 \leq 20, \\ \sigma/2, 20 < \sigma/2 < 40, \\ 40, \sigma/2 \geq 40. \end{cases} \tag{5}$$

Then, p can be classified by

$$U(p) = \begin{cases} 0, M(p) \geq T, \\ 1, M(p) < T, \end{cases} \tag{6}$$

where 0 and 1 indicate the background and foreground, respectively. T denotes the threshold of matching score.

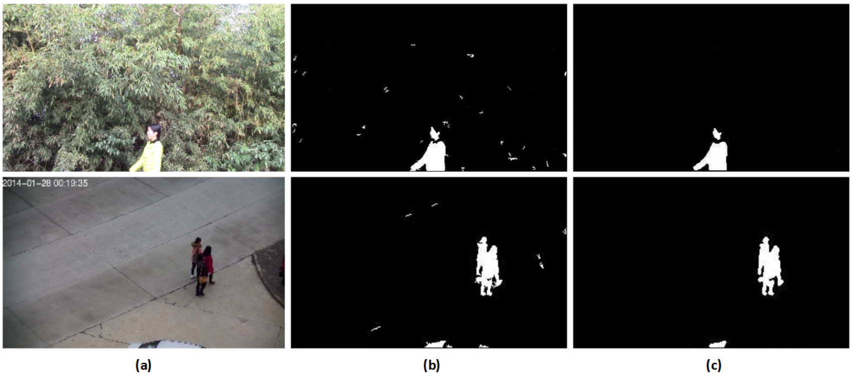

Fig. 2. Illustration of the noises produced by pixel classification and the results by morphological opening operation. (a) Denote the original frames, (b) Denote detection results with noises, and (c) Denote detection results post-processed by morphological opening operation.

3.4 Postprocessing

Due to the pixel-level modelling and classification, the proposed moving object detection may introduce some errors, which usually are isolated points. Therefore, the morphological opening operation is further utilized, in which the structural element is defined as 3×3, to remove these errors. Fig. 2 illustrates this process.

4 Experimental Result

In this section, our approach is evaluated on 10 collected challenging video sequences comparing with other state-of-the-art approaches, followed by the discussion of the efficiency analysis of our approach.

4.1 Evaluation Setting

The test videos are the real-life videos recorded from the university security monitoring system by PTZ cameras and hand-held cameras with resolution of 320×180 and frame rate 25fps. The evaluation is performed on 10 challenging video containing 4000 frames in total with vary moving objects, including pedestrians, cars, motorcycles and bicycles in dynamic background of the road or the playground, which take into account of the size and the type of moving objects as well as the camera movement and can comprehensively evaluate the performance of the proposed detection algorithm with others.

To make the comparison more comprehensive, the parameters are empirically fixed as $\{n, \eta, T\} = \{20, 0.2, 2\}$ in all evaluations.

4.2 Comparison Results

We compare our approach with two state-of-the-art moving object detection approaches, including DECOLOR [17] and ViBe [5]. Tab. 1 illustrates the average Recall (R), Precision (P) and F-measure on 10 collected video sequences while the detailed R and P values on each video sequence are shown in Fig. 3 and Fig. 4. We can conclude that our method can significantly outperforms other state-of-the-art in Precision and F-measure, although worse than others in Recall.

Table 1. The average R, P and F-measure values on 10 collected video sequences

	R	P	F-measure
DECOLOR	87.9%	49.1%	51.4%
ViBe	82.8%	22.7%	31.9%
Ours	70.1%	74.5%	70.6%

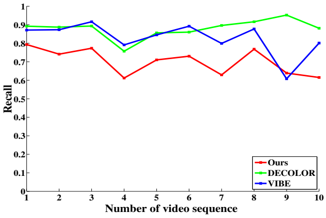

Fig. 3. Recall.

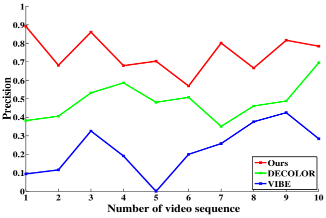

Fig. 4. Precision.

To demonstrate the performance of our proposed detection method against other two methods, we present some typical detection examples with different objects or backgrounds, as shown in Fig. 5. DECOLOR segments moving object in image sequence using a framework that detects the outliers to avoid complicated calculation, and uses low rank model to deal with complex background. It's easier to detect relatively dense and continuous region from the group. However, due to the smooth assumption of DECOLOR, more than one closed objects, especially in some occlusions, usually are detected as one single object (the second and third rows of the second column). ViBe produces ghost and has many noises (the second, the third and the fifth rows of the third column). From the comparative experiments, we can see that the proposed method outperforms DECOLOR in the details of objects, especially in the case of multiple objects, and is robust to the background interference compared with ViBe.

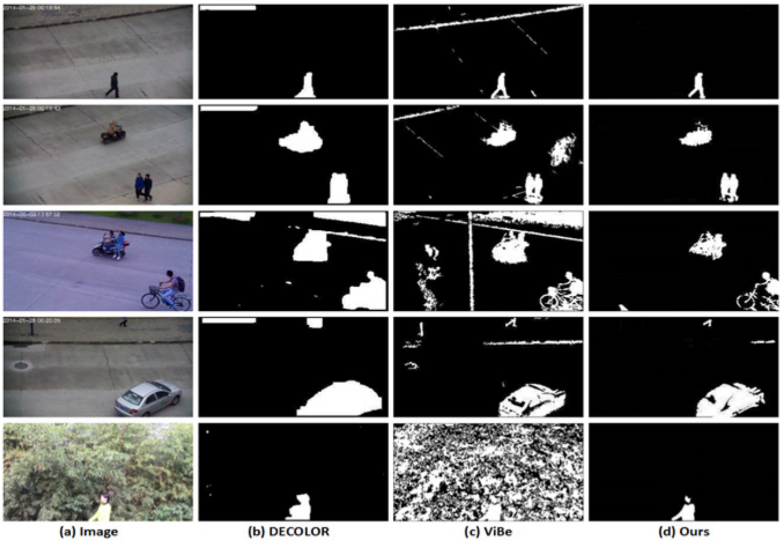

Fig. 5. Detection examples by our method comparing with other two methods, DECOLOR [17] and ViBe [5], with different objects under different dynamic backgrounds. The first column presents the sample frame of each type of video and the rest 3 columns present the detection results by DECOLOR, ViBe and the proposed method, respectively.

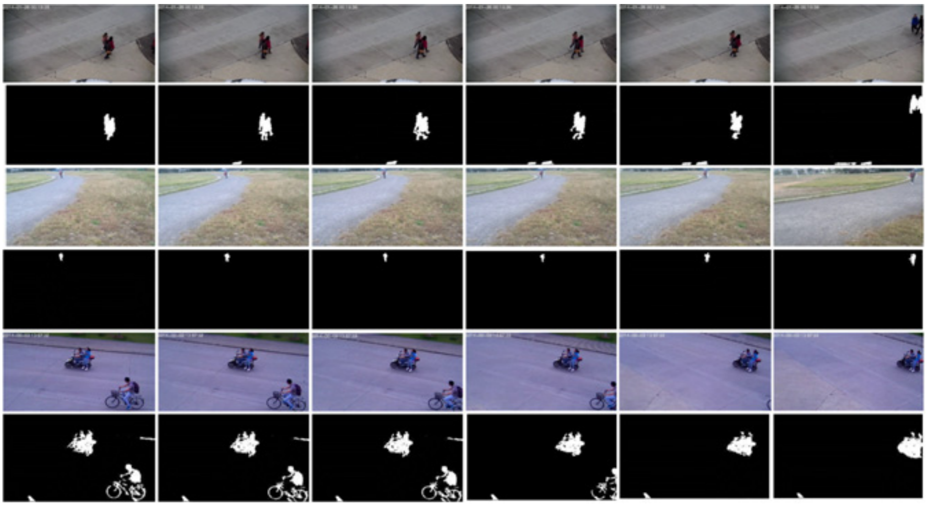

Fig. 6. The detection results of our proposed algorithm on 3 videos in every 5 frames. The 3 odd rows are the frames from 3 test videos respectively and the other 3 even rows are the corresponding detection results by our proposed method.

Fig. 6 presents the detection results of our proposed algorithm on every 5 frames of three videos. From Fig. 6 we can see that the proposed method can achieve superior performance in different surroundings with different types of objects in dynamic background.

4.3 Efficiency Analysis

The experiments are carried out on a desktop with an Intel i7 3.4GHz CPU and 32GB RAM, and implemented on C++ platform without any optimization. In the above experiments, the average runtime of proposed method is 0.12 second per frame while DECOLOR is 20 second per frame. Therefore, the proposed method are substantially faster than DECOLOR. In addition, our method is online while DECOLOR is a batch method. ViBe costs 0.02 second per frame, but it can only handle weak jitter problem of the camera, and is not suitable for the situation of dynamic background.

5 Conclusions

In view of the problems of moving object detection in dynamic background, this paper proposed a fast object detection method based on motion compensation. The background model of each pixel is initialized according to the first frame and is propagated to current frame by employing the edge-preserving optical flow algorithm to estimate the motion of each pixel. Each pixel can be finally classified as foreground or background pixel according to the compensated background model which is updated online by the fast random algorithm. The comparisons with DECOLOR and ViBe demonstrated the effectiveness of the proposed method, particularly in dynamic background. Moreover, the speed of proposed method achieved 8 fps. In future works, we will focus on developing more robust moving object detection approaches in real-time way to meet other applications.

Acknowledgments. Our thanks to the support from the National Nature Science Foundation of China (61502006, 61472002), the Natural Science Foundation of Anhui Higher Education Institutions of China (KJ2014A015) and the Natural Science Foundation of Anhui Province (1508085QF127).

References

1. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1), 1–31 (2011)
2. Bao, L., Song, Y., Yang, Q., Ahuja, N.: An edge-preserving filtering framework for visibility restoration. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 384–387. IEEE (2012)

3. Bao, L., Yang, Q., Jin, H.: Fast edge-preserving patchmatch for large displacement optical flow. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3534–3541. IEEE (2014)
4. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics-TOG **28**(3), 24 (2009)
5. Barnich, O., Van Droogenbroeck, M.: Vibe: a universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing **20**(6), 1709–1724 (2011)
6. He, K., Sun, J.: Computing nearest-neighbor fields via propagation-assisted kd-trees. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 111–118. IEEE (2012)
7. KaewTraKulPong, P., Bowden, R.: An improved adaptive background mixture model for real-time tracking with shadow detection. In: Video-Based Surveillance Systems, pp. 135–144. Springer (2002)
8. Lee, D.-S.: Effective gaussian mixture learning for video background subtraction. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(5), 827–832 (2005)
9. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3017–3024. IEEE (2011)
10. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1–3), 7–42 (2002)
11. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE (1999)
12. Szeliski, R.: Computer vision: algorithms and applications. Springer Science & Business Media (2010)
13. Tao, M., Bai, J., Kohli, P., Paris, S.: Simpleflow: a non-iterative, sublinear optical flow algorithm. In: Computer Graphics Forum, vol. 31, pp. 345–353. Wiley Online Library (2012)
14. Van Droogenbroeck, M., Paquot, O.: Background subtraction: experiments and improvements for vibe. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 32–37. IEEE (2012)
15. Yang, Q., Yang, R., Davis, J., Nistér, D.: Spatial-depth super resolution for range images. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2007, pp. 1–8. IEEE (2007)
16. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4), 650–656 (2006)
17. Zhou, X., Yang, C., Yu, W.: Moving object detection by detecting contiguous outliers in the low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(3), 597–610 (2013)
18. 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)