A novel object detection technique for dynamic scene and static object

Xiu Li, Liansheng Chen, Zhixiong Yang and Huimin Wang

Shenzhen Key Laboratory of Information Science and Technology Graduate School at Shenzhen, Tsinghua University Shenzhen, China

Abstract. The object detection in video streams plays an important role in computer vision applications. The background subtraction, comparing each new frame to a model of the background, is one of the most popular method. However, the static background is practically impossible, and dynamic background makes the perfect object detection difficult. In addition, the problem of static object significantly affects the performance. In the paper, based on the visual background extraction (ViBe) algorithm, we presented a new method to deal with all problems. For the dynamic scene, we presented a new update mechanism to obtain the more robust model in dynamic regions. To address the static object issue, we cancelled the propagation mechanism in ViBe, designed an algorithm to detect region where it is always detected as foreground, and distinguished the static object from ghost with a self-designed measurement, which combined the knowledge of region contrast and edge. We described our method in full details, and compared it with other background subtraction techniques. Our experimental results show that the proposed method outperforms several state-of-the-art object detection approaches. In addition, it can process 60 frames per second on an Intel i5 3.1 GHz CPU with C++ code.

1 Introduction

In image processing and computer vision applications, accurate object extraction from the video stream is an important precursor for consecutive tracking, recognition and behaviour analysis. Background subtraction, comparing an observed image with the preserved background model, has become the mainstream technique for accurate foreground extraction.

In the simplest case, a static background frame is compared with the current frame, and pixels with high deviation are classified as foreground. However, it is rarely the case, and the stationary camera does not mean a static background because of complicated environments, such as water wave, spring and swaying leaves, et al. [1], which trigger many false alarms. The camera jitter [2] makes the situation worse.

In addition, there are problems which have not been paid enough attention but significantly affect the performance, intermittent object [1], ghost and bootstrapping [2]. When the intermittent object rests, and the model is updated blindly [3], the foreground object will be gradually incorporated into model and removed from the detection result. If the object moves again, both it and the newly revealed parts of the background, called ghost, are detected. If the model is updated conservatively [3], everything seems to be OK. Actually, it is not the case. The problem of bootstrapping makes the initial background model polluted by some foreground information inevitably. The conservative update mechanism will lead to ghost when an object initially in the background model moves. The problem of bootstrapping is a special case for problems of intermittent object and ghost. No existing method can remove the ghost, detect the intermittent object and solve the problem of bootstrapping simultaneously. We call these three problems as static object because all negative effects result from the static behaviour of the object.

Both the phenomena of dynamic scene and static object are common. Dynamic scene, such as water wave, spring and swaying leaves, et al., is inevitable in the natural environment. People and cars at the traffic lights are typically static object. Both of them introduce negative effects for the accurate object detection. Based on visual background extraction (ViBe) [3], we presented a new method, which can deal with all problems mentioned above.

The movement of dynamic background leads to the pixel value failing to be compared with the corresponding model, which results in false alarms. However, the displacement in consecutive frames is small, and thus corresponding model can be found in a small neighbourhood, which named as the neighbourhood match. If there exists a successful match, we will replace two samples in the model with the value, which called sharp update. If these two operations are conducted for all pixels, there will be many false negatives and foreground information will pollute the background model. Thus, these operations just worked in the dynamic background region.

To solve the problem of intermittent object, ghost and bootstrapping, we present a unified framework. First, we cancel the propagation mechanism [3], and update the model conservatively. In this way, ghost and static object will be detected as foreground all the time. As a result there will be some regions, where are always detected as foreground. Then, we design an algorithm to detect these regions. Finally, a new measurement is designed to distinguish ghost from the static object, based on which the model in the region of ghost is modified and the previous results are corrected. Thus, the ghost will be removed, and the static object will be detected successfully.

The rest of this paper is organized as follows. In section 2, we present related background subtraction methods. In section 3, a detailed description of our algorithm is provided. Then, the experimental results of the proposed method in several videos are given in Section 4. Finally, we summarize this paper and suggest possible future work in Section 5.

2 Related Work

Background subtraction has become the mainstream technique in object detection. The main idea is to create and maintain a model of the scene without objects, and detect the foreground by comparing the current frame with the estimated background. The advantage of this concept is that no prior knowledge is required to detect the object as long as their appearance differs enough from the background (i.e. they are not camouflaged [2]). A multitude of algorithms and methods for background modelling have been developed, such as the Gaussian mixture model (GMM) [4], kernel density estimate (KDE) [5], Codebook (CB) [6], Visual background Extraction (ViBe) [3] and many other improved versions [7][8][9][10] based on these methods. Excellent survey papers can be found in [2][11][12][13].

Because of the dynamic nature of real-world scenes, there are inevitably many false alarms. Numerous methods have been presented to handle with this issue. Some statistical models [4][5] are used to represent the multi-modal essence of dynamic background. In fact, these models are only able to represent very small background movement. Many people hold the idea the model converges too slowly, and designed some algorithms [9][14] to speed up the update process in the dynamic region. However, there is still some delay in the model. Some leave the problem to be solved in regularization step [7].

Static object is another problem in background subtraction. If the model is updated blindly, ghost and static object will be incorporated in the model. If the model is updated conservatively, ghost and static object will be detected forever. It is a difficult task to detect static object all the time and remove ghost in no time. Some works have been done to tackle with this problem. Taycher [15] firstly proposed the statistically consistent method for incorporating feedback from the high-level motion model to modify adaptation behaviour, which depends heavily on an accurate tracker. Wang and Bunyak [16] fused flux tensor and split Gaussian based on detection results to get the region of static object and ghost, and exploited edge knowledge in these regions to distinguish them. It detects static foreground and removes ghost successfully, but fails to run in real time.

3 The Proposed Method

Based on ViBe [3], we propose a new approach to handle the problem of dynamic scene and static object. For the dynamic scene, we present a mechanism of the neighbourhood match and sharp update to remove false alarms and speed up update process in the dynamic region, which is described in Section 3.1. For static object, we design an algorithm to find out the regions, where is always detected as foreground. Then, we figure out the objects covering the regions by clustering the foreground pixels in these regions with the sequential leader clustering (SLC) algorithm [17]. Finally, we distinguish static object from ghost by a self-designed measurement with the knowledge of region contrast [18] and edge. Detailed description is provided in Section 3.2. An overall framework is shown in Figure 1.

3.1. Dynamic Scene Noise Removal

In outdoor environments with fluctuating background, the false alarms are derived from two sources. First, there are false alarms because of random noise, which should be homogeneous over the entire image. Second, there are false alarms because of small movements in the scene background that are not represented in the background model. For example, a tree branch moves further than it did during model generation. In addition, small camera displacements because of wind load are common in outdoor surveillance and trigger many false alarms. This false detection is usually spatially clustered in the image, which is not easy to eliminate using morphology or noise filtering. Because these operations might also affect small and occluded targets. This method aims to suppress the false detections due to small and unmodelled movements in the scene background.

In ViBe, the pixels are compared with their own models and the ones similar with at least two samples in the model considered as background, having a chance of one sixteenth to make its model and the model of neighbourhood updated with its current value. In this way, only pixels sharing similar appearance with the samples in the model can be incorporated into the model. Thus, the model just contains samples from one object in most cases. Although there is a probability the model contains samples from different objects because two objects are similar with the same object may be different, it takes a long time to collect enough samples and requires special condition the pixel values vary slowly, such as slow illumination. In addition, with the mechanism of propagation, updating the model of its neighbourhood with its own pixel value, the model in the edge of objects may have the chance of containing samples from different objects, but collecting enough samples takes a long time. The models in the region of dynamic

DOI: 10.1051/matecconf/20165408003

background should always be multi-modal. Thus, the algorithm of ViBe fails to handle the problem of the dynamic scene. The false alarms because of the small

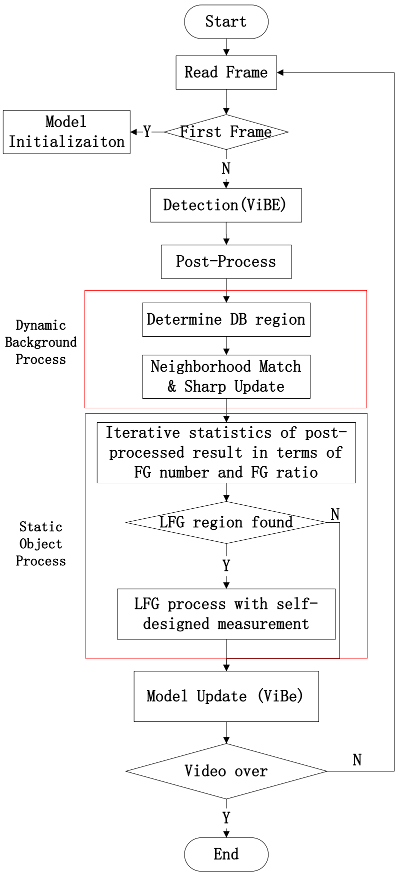

Figure 1. Overall framework of our method. DB is short for Dynamic Background and LFG is short for Long-time Foreground.

movements of background objects prevent the moved background objects incorporating into the models of new pixel locations, and the unchanged models will cause similar false alarms, which is a drop-dead halt.

Parts of the background move to occupy a new pixel, but it was not part of that pixel, then it will be detected as a foreground object. However, this object will have a high probability to be a part of the background distribution at its original pixel. If only a small displacement occurs between consecutive frames, we judge whether a detected pixel is caused by a background object that has moved by comparing the detected pixel value with models in the neighbourhood and finding out some matched models. We name the procedure of finding matched models in the neighbourhood as neighbourhood match, which can break the drop-dead halt.

The object of dynamic background does not stay too long on one spot. Collecting enough samples (typically 2 samples) to represent the moving object needs about the time of 32 frames in the situation that the object is there. If the object is always moving, it will take more time to collect enough samples. Moreover, if enough samples fail to be collected immediately, some false alarms of background objects will also fail to be found out and removed, which further prevent the background objects incorporating into the models of new pixel locations. Although the mechanism of neighbourhood match breaks the drop-dead halt, the slow update mechanism leads to a vicious circle. To break the vicious circle for the false alarms of moving background object, we replace two samples in the model of new pixel location with two copies of the current pixel value, which is named as sharp update. In this way, the moving background objects can be incorporated into the models of new pixel locations immediately, and further movements can also be detected. Thus, a more robust and reliable model is obtained, and fewer false alarms will be detected, which is beneficial to post process, such as median filtering and morphological operations.

In addition, operations of the neighbourhood match and sharp update are only conducted in the region of dynamic background, which is indicated in a 2D map of pixel level, named as blink map. In [19], for each pixel, the previous updating mask (prior to any modification) and a map with the blinking level are stored. This level is determined as follows. For a pixel, if the current updating label is different from the previous one, then the blinking level is increased by 15 (the blinking level being kept within the [0,150] interval), while the level is decreased by 1. A pixel is considered as blinking if its level is larger or equal to 30. Directly detecting blinking pixels following the steps above is inconvenient, since the borders of moving foreground objects would also be included in the result. Thus, we find blinking pixels just in the region of background, which is indicated in the post-processed and dilated updating mask. As it is shown in Figure 2, we presented several results from ViBe and our method, and much fewer false alarms are detected in our results, which demonstrates the advantage of our model.

3.2 Static Object Detection

The static objects (true positives) and revealed background by removed objects (false positives) called as ghost are detected as foreground by traditional method, such as ViBe. If the model is updated blindly, ghost and static object will be incorporated into the model. If the model is updated conservatively, ghost and static object will be detected forever. In fact, the ghost should be included into model, and the static object should be detected as foreground. It is hard to distinguish them from each other just with the temporal knowledge. Thus, we should exploit some spatial knowledge. The static objects are different from the surrounding background while the revealed background by removed objects is similar with the surrounding background, and our method is designed based on it. Detailed descriptions are given as follows.

First, we choose the conservative update mechanism and cancel the spatial propagation mechanism [3] in ViBe in order to make the static objects, and revealed background by removed objects detected as foreground all the time. Then, we find out them by searching for regions, it is always detected as foreground. Considering the factors of noise, illumination change, occlusion, et al, the static objects and the revealed background by

DOI: 10.1051/matecconf/20165408003

removed objects are not detected as foreground continuously. Thus, a more reliable approach is designed.

The main idea is where that if a pixel is detected as foreground frequently enough in a certain period, it is considered as a long-time foreground (LFG, for short) pixel. Specifically, if in a period, a pixel is detected as foreground more than 500 times and the ratio of foreground is more than 80%, we consider it as a LFG pixel. In addition, the pixels in a LFG object are detected at different time, but the time interval is not big. In order to detect the complete LFG, we stop detecting new LFG pixels when no new LFG pixel is detected for 100 frames. Furthermore, these foreground pixels are stored and will be used to figure out the object covering the region of long-time object with the algorithm of SLC [17].

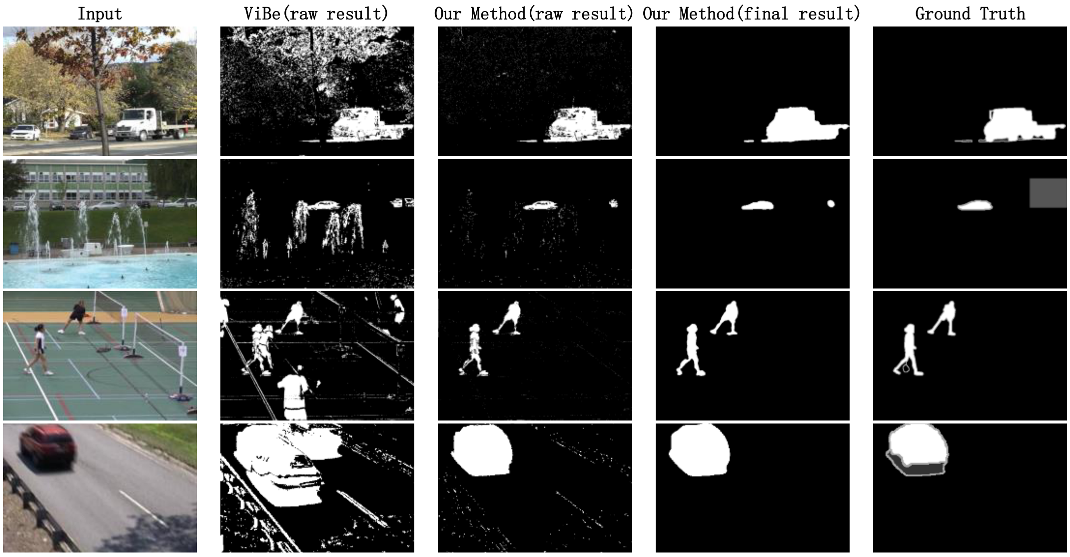

Figure 2. Comparison with ViBe (the threshold R = 10)

Finally, we design a measurement to distinguish the static object and ghost. The concept of region contrast given in [11] can be used to measure the similarity between the central and surrounding region. However, it is difficult to set a global threshold for all cases. Thus, we compute the region contrast between the central and surrounding model, RC_M, and the region contrast between the central cluster result and surrounding model, RC_C, respectively.

$$RC\_M = (\sum_{p_i \in P_{MC}} \sum_{p_j \in P_{MS}} D(p_i, p_j)) \div (\# P_{MC} \times \# P_{MS}) \quad (1)$$

$$RC\_C = (\sum_{p_i \in P_{CC}} \sum_{p_j \in P_{MS}} D(p_i, p_j)) \div (\# P_{CC} \times \# P_{MS}) \quad (2)$$

where $D(p_i, p_j)$ is the color Euclidean distance metric between pixels p_i and p_j in the R*G*B space, P_{MC} stands for the pixels in the model of the center region of a long-time foreground object, the white region in Fig 3.b, P_{MS} stands for the pixels in the model of the surrounding region of a long-time foreground object, the gray region in Fig 3.b, P_{CC} stands for the pixels in the cluster result of the center region of a long-time foreground object, $\# P_{MC}$, $\# P_{MS}$, $\# P_{CC}$ are the number of pixels in P_{MC}, P_{MS}, P_{CC}.

If RC_M > 1.5*RC_C, the model is modified with the cluster result. Illustrations are given in Fig 3.

However, it only works when the background is flat. To solve this problem, we further exploit the knowledge of edge inspired by the idea given in [16]. Specifically, we measure the degree of flatness with variance and compute the variance for center model (VAR_M) and center cluster (VAR_C) in the LFG region, respectively.

$$VAR\_M = (\sum_{p_i \in P_{MC}} D(p_i, \overline{p_{MC}})) \div (\# P_{MC}) \quad (3)$$

$$VAR\_C = (\sum_{p_i \in P_{CC}} D(p_i, \overline{p_{CC}})) \div (\# P_{CC}) \quad (4)$$

where $\overline{p_{MC}}$ and $\overline{p_{CC}}$ are the average pixels in P_{MC} and P_{CC}.

And if 0.66*RC_C<RC_M<1.5*RC_C and max(VAR_M, VAR_C)>8000, we follow the steps in [16] to respectively figure out the edges in model image, cluster image and binary image indicating the LFG object, and modify the model when the edges in binary image are more similar with the ones in the model image. A visual and intuitionistic flow chat is given is Fig 4.

In addition, when some LFG objects are spatially close, these objects are detected as a combined LFG object due to some morphological operations in our method, which is not beneficial for the distinguishing procedure. In order to handle the problem, we further exploit the knowledge of the starting time of LFG object. In details, we split the combined object into several parts, each of which just belongs to a real LFG object, with analysis of histogram of the starting time of LFG object and deal with each part as mentioned above. The main idea of the histogram analysis is to find some isolated intervals, whose histogram value is larger than certain threshold. Specifically, we first generate a histogram of ts with the bin width being about 100. To find the left border of each interval, we look for the first bin, whose histogram value is larger than 0.005, from the right border of the previous interval and set the left border of the found bin as the left border of a new interval. To find the corresponding right border, we look for the first bin, whose histogram value is less than 0.005 from the corresponding left border and set the left border of the found bin as the corresponding right border of the new interval. We present a concrete case in Figure 5.

The modified model will be used for the correction of previous wrong result and object detection in following frames. Furthermore, we find that our method can deal with the problem of bootstrapping in Figure 2.

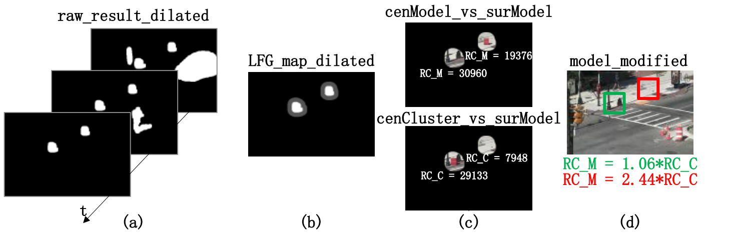

Figure 3. These images are all chosen from the video of abandonedBox (a) dilated version of raw result, (b) obtained map of LFG region (the center white region is LFG region, and the surrounding gray region is obtained with dilation operation) (c) comparative image (in the up image the center region is model and the surrounding region is model as well, and in the down image the center region is the cluster result using SLC and the surrounding region is model) used to compute the measurement of region contrast, (d) modified model image

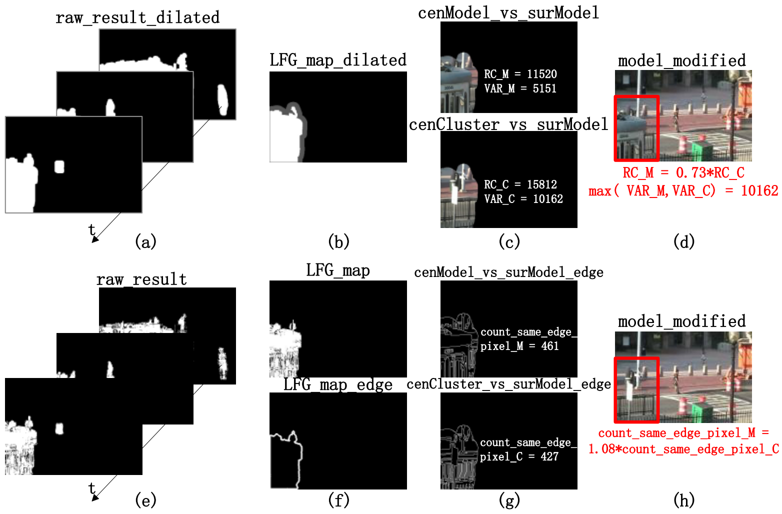

Figure 4. These images are all chosen from the video of tramstop (a) dilated version of raw result, (b) obtained map of LFG region from dilated results, (c) comparative image (VAR_X is the variance of the center region), (d) modified model image (the model is not modified successfully with the knowledge of region contrast but 0.66*RC_C<RC_M<1.5*RC_C and max(VAR_M, VAR_C)>8000), (e) raw result, (f) obtained map of LFG region from raw result and obtained edge from it, (g) obtained edge from (c), (h) modified model image

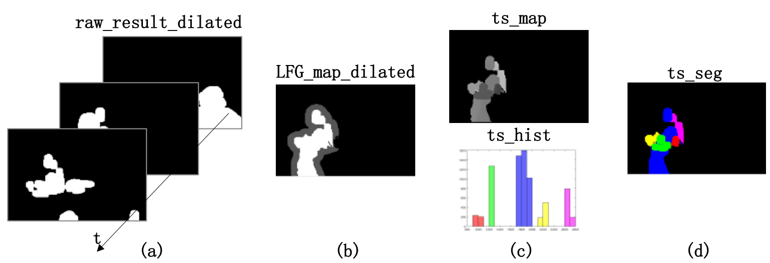

Figure 5. These images are all chosen from the video of diningRoom (a) dilated version of raw result, (b) obtained map of LFG region from dilated results, (c) map of the starting time of LFG object and the histogram of it, (d) the segmentation of ts_map from the analysis of its histogram

4 Experimental Results

The proposed method is evaluated using the dataset and evaluation metrics in CVPR 2012 Change Detection challenge (CDnet 2012) [1].A fixed set of parameters is used for all sequences, and some of them inherit from [3][19]. All experiments run on an Intel i5-3450 processor with 3.10GHz, 16GB RAM and Win7 OS. The algorithm is implemented by C++ codes, and the average processing speed is about 60 fps for the image size of 320x240.

4.1. Performance for dynamic scene and static object

In the CDnet 2012 dataset, the video sequences of dynamic background (DB) and camera jitter (CJ) share the problem of dynamic scene, and the ones of intermittent object (IO) share the problem of static object. Some of these video sequences share the problems of shadow, highlight, illumination change and camouflage. To demonstrate the contributions of our method, we compare it with several state-of-the-art methods in the video sequences of badminton in CJ, canoe, fall, fountain01 and overpass in DB, and tramstop in IO, which do not share the problems of shadow, highlight, illumination change and camouflage. These state-of-the-art methods include ViBe [3] on which our method is based; SuBSENSE [9] is the best one in all improved versions for ViBe; CDet [20] performs best for the video sequences in DB; GPRMF [20] performs best for the video sequences in CJ; and FTSG [16] performs best for the video sequences in IO. Detailed statistical data is provided in Table I and the data in bold is the best in each row. Obviously, our method performs better than the other five methods for dealing with the problem of dynamic scene and static object. For the dynamic scene, our method is better than the other methods on most of the video sequences, and a little worse than the best one is in a few of video sequences. The overall measurement of our method is better than that of the other methods. Although our performance is not the best for static object, the gap between the best one and our method is small. In addition, FTSG fails to run in real time, while our method succeeds and performs much better than FTSG for the dynamic scene. Finally, for both dynamic scene and static object, the overall performance of our method is better than that of the other methods.

4.2. Overall performance and discussion

We compared our method with methods mentioned in section 4.1 using the metrics of recall, precision, F-Measure and time cost. Table II shows the statistical data. Our method performs much better than ViBe, and increases the first three metrics by more than 10%. Although SuBSENSE, CDet and FTSG obtain better performance, the gaps are not large. These three methods design some tricks to deal with the problem of shadow, highlight, illumination change and camouflage, which are the reasons of these gaps. However, these tricks introduce some extra time cost. Thus, our method runs faster than the other three methods.

5 Conclusions

In this paper, we presented a ViBe-based background subtraction algorithm, which deals with the problem of dynamic scene and static object well. Neighbourhood match can remove the false alarms of dynamic scene, and sharp update can speed up the update process, which make the model more robust and reliable. A novel algorithm is presented to detect the LFG region, and a new measurement is designed to distinguish static object from ghost. Experiments show they work well together. As mentioned above, no specific mechanisms are designed to deal with the shadow, highlight, illumination change and camouflage in our algorithm. We believe a good representation of the pixel will be the answer, and try to find out the appropriate representation to deal with these issues in the future.

Acknowledgment

This research was partly supported by National Natural Science Foundation of China (NSFC, Project No.: 71171121/61033005) and National "863" High Technology Research and Development Program of China. (863 Project No.: 2012AA09A408).

Table 1. Method Comparison on performance for dynamic scene and static object

category	video name	measure \ method	ViBe	SuBSENSE	CDet	GPRMF	FTSG	Our Method
dynamic scene	badminton	recall	0.799	0.922	**0.971**	0.823	0.917	0.965
		precision	0.576	0.843	0.851	**0.964**	0.916	0.937
		F-Measure	0.669	0.881	0.907	0.888	0.916	**0.951**
	canoe	recall	0.858	0.659	**0.966**	0.945	0.913	0.950
		precision	0.855	**0.993**	0.974	0.978	0.985	0.957
		F-Measure	0.856	0.792	**0.970**	0.961	0.948	0.953
	fall	recall	0.789	0.857	**0.998**	0.969	0.988	0.988
		precision	0.267	0.876	0.887	0.199	0.878	**0.942**
		F-Measure	0.399	0.866	0.926	0.331	0.930	**0.964**
	fountain02	recall	0.802	0.923	**0.950**	0.884	0.947	0.945
		precision	0.862	**0.966**	0.948	0.937	0.955	0.965
		F-Measure	0.831	0.944	0.949	0.910	0.951	**0.955**
	overpass	recall	0.763	0.785	0.882	**0.980**	0.944	**0.980**
		precision	0.851	0.943	0.919	0.891	0.941	**0.970**
		F-Measure	0.805	0.857	0.900	0.934	0.943	**0.975**
static object	tramstop	recall	0.405	0.417	0.972	0.734	**0.987**	0.981
		precision	0.272	0.831	**0.976**	0.1934	0.967	0.964
		F-Measure	0.325	0.555	0.974	0.306	**0.977**	0.972
overall	overall	recall	0.736	0.761	0.957	0.889	0.949	**0.968**
		precision	0.614	0.909	0.926	0.694	0.940	**0.956**
		F-Measure	0.648	0.816	0.937	0.722	0.944	**0.962**

Table 2. Method comparison on overall performance

	ViBe	SuBSENSE	CDet	GPRMF	FTSG	Our Method
recall	0.682	0.828	**0.903**	0.837	0.838	0.846
precision	0.736	0.857	0.840	0.814	**0.868**	0.854
F-Measure	0.668	0.826	**0.861**	0.794	0.835	0.818
time cost (ms/frame)	**2 (C++) i5 3.1GHz**	22 (C++) i5 3.3GHz	25 (C++) 2.8GHz	unknown	100 (Matlab) Intel 2.4GHz	16 (C++) i5 3.1GHz

References

1. Goyette, Nil, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. "Changedetection. net: A new change detection benchmark dataset." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 1-8. IEEE, 2012.
2. Bouwmans, Thierry. "Traditional and recent approaches in background modeling for foreground detection: An overview." *Computer Science Review*11 (2014): 31-66.
3. Barnich, Olivier, and Marc Van Droogenbroeck. "ViBe: A universal background subtraction algorithm for video sequences." *Image Processing, IEEE Transactions on* 20, no. 6 (2011): 1709-1724.
4. Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.
5. Elgammal, Ahmed, David Harwood, and Larry Davis. "Non-parametric model for background subtraction." In *Computer Vision—ECCV 2000*, pp. 751-767. Springer Berlin Heidelberg, 2000.
6. Kim, Kyungnam, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. "Real-time foreground–background segmentation using codebook model." *Real-time imaging* 11, no. 3 (2005): 172-185.
7. Sheikh, Yaser, and Mubarak Shah. "Bayesian modeling of dynamic scenes for object detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 11 (2005): 1778-1792.
8. St-Charles, Pierre-Luc, Guillaume-Alexandre Bilodeau, and Robert Bergevin. "A Self-Adjusting Approach to Change Detection Based on Background Word Consensus." In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pp. 990-997. IEEE, 2015.
9. St-Charles, Pierre-Luc, Guillaume-Alexandre Bilodeau, and Robert Bergevin. "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity." *Image Processing, IEEE Transactions on* 24, no. 1 (2015): 359-373.
10. Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 28-31. IEEE, 2004.
11. Brutzer, Sebastian, Benjamin Höferlin, and Gunther Heidemann. "Evaluation of background subtraction techniques for video surveillance." In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1937-1944. IEEE, 2011.
12. Parks, Donovan H., and Sidney S. Fels. "Evaluation of background subtraction algorithms with post-processing." In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pp. 192-199. IEEE, 2008.
13. Radke, Richard J., Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. "Image change detection algorithms: a systematic survey." *Image Processing, IEEE Transactions on* 14, no. 3 (2005): 294-307.
14. Hofmann, Martin, Philipp Tiefenbacher, and Gerhard Rigoll. "Background segmentation with feedback: The pixel-based adaptive segmenter." In*Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 38-43. IEEE, 2012.
15. Taycher, Leonid, John W. Fisher III, and Trevor Darrell. "Incorporating object tracking feedback into background maintenance framework." In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, vol. 2, pp. 120-125. IEEE, 2005.
16. Wang, Rui, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan. "Static and moving object detection using flux tensor with split Gaussian models." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pp. 420-424. IEEE, 2014.
17. Hartigan, John A. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
18. Cheng, Ming, Niloy J. Mitra, Xumin Huang, Philip HS Torr, and Song Hu. "Global contrast based salient region detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 37, no. 3 (2015): 569-582.
19. Van Droogenbroeck, Marc, and Olivier Paquot. "Background subtraction: Experiments and improvements for ViBe." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 32-37. IEEE, 2012.
20. changedetection website. http://changedetection.net/.