MCSHM: A Simple and Practical Method for Moving Objects Detection in Dynamic Scenes*

Li Jinping[1,2], Guo Yukun[1,2], and Ma Liwen[1,2]

1. School of Information Science and Engineering, University of Jinan, Jinan 250022, China
2. Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China

***Abstract*—Moving objects' detection in dynamic scenes is a very important task in video processing. In the applications of image processing (for example, video-surveillance), more attention is paid to whether there is an interested object in the scene rather than where the object is located. As a matter of fact, similar to static backgrounds, the statistical histograms of the most dynamic backgrounds have favorable stability, and these histograms would change clearly only when big and contrasting objects enter or move out of the scenes. This is a very important and interesting property for dynamic background. We propose a fast and simple algorithm, combining histograms in multiple color spaces and the superposition principle of statistical histogram, called Multiple Color Space Histogram Models (MCSHM). MCSHM first calculates statistical histograms of many color components in multiple color space and then use the changes of statistical histograms to determine whether there is an object rather than the changes in pixels or pixel-level regions. Thus, the computational complexity of MCSHM is kept at a very low level. The basic steps are as follows: firstly, convert each frame from RGB space to other color spaces and calculate the histograms of selected color components, then we can obtain the background histogram model; secondly, detect the objects using statistical histogram superposition principle; finally, update MCSHM by the result of detection. The experimental results demonstrate that our method can quickly and accurately detect moving objects in dynamic scenes.**

Keywords*—*dynamic scene*; o*bject detection*; *Multiple Color Space Histogram Models*; *superposition principle

I. INTRODUCTION

Dynamic scenes include two basic categories: one is caused by the motion of the camera due to vibration, rotation, translation and zoom. The other refers to the fixed camera with changing background, such as trees and water surface in the wind. In the current paper, we deal with the latter case. In general, the majority of the security surveillance videos belong to the latter. The purpose of this paper is to construct a fast, simple and robust algorithm for detecting moving objects in dynamic scenes.

A. Related works

The simple detection method (comparing a static background frame with the current frame, pixel by pixel) cannot update the background model and the corresponding detection error increases over time, so many researchers tried to explore new methods.

In order to improve the adaptability of illumination change in the background, Ridder et al. [1] modeled each pixel with a Kalman filter. This method can well handle the background illumination change, but the background recovers very slowly after objects move out of the scene. Pfinder [2] is a system for tracking people and interpreting their behavior. To segment the human silhouette, it models each pixel with a Gaussian model. In this way, Pfinder can effectively track the human body in the indoor environment but this method does not mention the effectiveness in the outdoor case. Gaussian Mixture Model (GMM) proposed by Stauffer et al. [3,4] to improve the single Gaussian model is probably the most popular algorithm in modeling background. In GMM, pixel intensity is generally modeled by a mixture of K Gaussian distributions (typically, K is within 3 to 5). This model can update the parameters and weights online. So, GMM can address the problems of illumination changes, leaves swaying and water ripples etc. Since then, many scholars have put forward various improved GMM algorithms [5–11] and obtained favorable experimental results. One disadvantage of GMM is that the modeling process is too complex (each pixel should go through a complex calculation to determine which part it belongs to), furthermore, the parameters estimation in GMM is susceptible to noise. Shoushtarian et al. [12] proposed and compared three kinds of dynamic scenes extraction algorithms, Selective Update using Temporal Averaging, Selective Update using Non-Foreground Pixels of the Input Image and Selective Update using Temporal Median, the experimental results showed that the third algorithm has the best performance among the three algorithms, and it works well in unconstrained outdoor and indoor scenes.

Zhang et al. [13] proposed an algorithm for dynamic scene subtraction based on a covariance-based method (CM). CM combines two distinct levels: pixel level and regional level. At the pixel level, coordinate values, intensity, texture and gradient etc. are used as features of each pixel. At the regional level, the features that are extracted at the pixel level are represented by a covariance matrix, which is calculated over a rectangle region. In CM, each pixel is modeled as a group of weighted covariance matrices. The experimental results show that CM can handle dynamic scenes very well while its high computational complexity causes low efficiency. Zhang et al. also proposed Local Dependency Histogram (LDH)

*Supported by Shandong Provincial Key Project (2015ZDXX0801A03) and Shandong Scientific Projects in Universities (J14LNXX)。

model[14], and obtained good experimental results. Barnich et al. put forward a novel method to estimate the background, ViBe [15,16]. This is the first time that a random aggregation is used in the field of background extraction, and the authors proposed a novel policy that propagates information between neighboring pixels of an image. The method is said to be highly robust to noisy images. However, its ability to cope with background of a dynamic (tree leafs, water fluctuation, etc.) is debatable. Droogenbroeck et al. presented numerous modifications of the original ViBe algorithm, ViBe+ [17]. Experimental results show that ViBe+ is preferable as compared to the original ViBe version. Chen et al. [18] improved ViBe in the suspects of initialization, matching and the update mechanism of ViBe. These improvements increase the detection accuracy of ViBe in dynamic scenes. Hofmann et al. proposed a non-parametric moving object detection algorithm, Pixel-Based Adaptive Segmenter (PBAS) [19], which combined the advantages of SAmple CONsensus (SACON) [20] and ViBe. PBAS can adjust the threshold and learning rate adaptively depending on the complexity of the background. So its ability to adapt to the background is very strong. However, the computational complexity of this algorithm is insufferable. Yoshinaga et al. [21] put forward two types of spatiotemporal background modeling frameworks, Statistical Local Difference Pattern (SLDP) and Spatiotemporal Similarity of Intensity Changes (StSIC). These two methods use the spatial information to absorb the effects of illumination changes and use the temporal information to handle the dynamic changes. Sobral et al. [22] tested 29 background subtraction algorithms on the BMC data set. They conducted a relevant experimental analysis to evaluate both the robustness and practical performance in terms of processor and memory requirements.

B. Motivation and Our method

For video-surveillance, people just want to know if there is some motion in the scene but the location of the object is out of consideration. We proposed a novel algorithm to detect if there are some moving objects in the scene.

Statistical histogram is a very commonly used tool in image processing but it discards all the spatial information of pixels. Spatial information loss will affect the location of the object. While for the video-surveillance, that does not need spatial information, this characteristic of statistical histogram can eliminate the interference caused by spatial information instead. In this current paper, we don't care about the location of the object but only wish to determine whether a moving object appears in the scene. So, using the spatial-information-insensitive characteristics of statistical histogram to model the background is an appropriate strategy.

For most dynamic scenes, the inherent attributes (intensity, texture, topology structure, etc.) of dynamic scenes are stable although the background is constantly changing. As a matter of fact, the inherent attribute that histogram describes is intensity. Statistical histogram will show a very good stability when there is no object in the scene; if an object appears, the histogram will have an obvious change. As shown in Fig.1, the histograms (hue, saturation and value) are stable when there is no object. In the 2637th frame, all the histograms have obvious fluctuations caused by the man's appearance. After the man left, all the histograms return to the previous states.

In order to describe the background histogram model more accurately, we adopt multiple color space (introduced in section II) because of its good complement and redundancy. Multiple color components histograms can retain more information in the image. So, it can describe the background model better than a single color component. In addition, we use histogram superposition principle (introduced in section III) to capture the changes in histograms. In multiple color space, we utilize the range of each color component's histogram to determine if an object has appeared. The final step is updating the background model of the learning factor. Due to the fact that our method is histogram-level modeling rather than pixel-level modeling, therefore, our method has simpler calculations and better feasible.

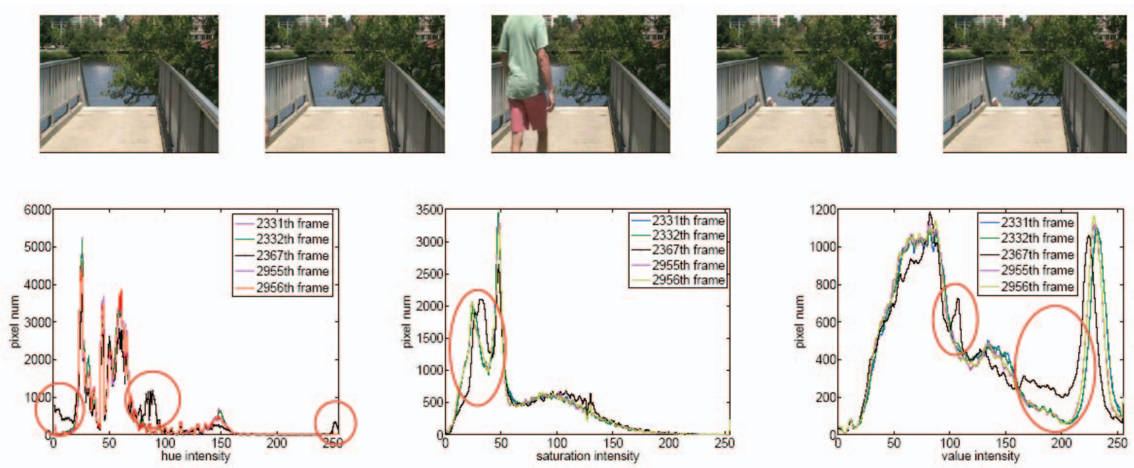


Fig.1 The moving object can affect the histograms. First row: Video frames: 2331th, 2332th, 2367th, 2955th and 2956th in "overpass"[23]. Second row: histograms of hue, saturation and value.

II. MULTIPLE COLOR SPACE

In order to adapt to different application scenarios, scholars put forward many color spaces such as HSV, YUV, RGB, HIS and Lab etc. Researchers can choose an appropriate color space to describe the characteristics of objects in different application condition according to the characteristics of different color spaces. The different components of a color space are not related so each color can be uniquely expressed in terms of the value of each component of a color space. But the components of different color spaces are not necessarily independent.

Consider a situation in which we want to describe the color feature of an object in a color image. The direct way is to extract the color values of the pixels contained in the object. Although we can easily get the R, G and B values of the pixels in RGB color space, to get the saturation and hue of the color in RGB color space is impossible. To obtain this information, we need to map the image from RGB color space to HSV color space. Furthermore, we can map the image to other color spaces to extract more color features. In this paper, the multiple color space is simply described as follows: mapping image to different standard color spaces, and extracting the color components into a new color space. This new color space is called multiple color space. Multiple color space pays attention to the complementarity between the different color components, and does not emphasize on its completeness and redundancy of itself.

For example, if we want to extract a certain color from an image. We can merge the color components of R, G, B, H and S into one color space to extract the color information that we are interested in. Additionally, we can avoid choosing the component which is useless even disruptive for the feature extraction, for instance, removing the V component in the HSV color space to reduce the effect of illumination changes. So the multiple color space is very flexible. It can combine any color components as needed to achieve the desired goal.

III. HISTOGRAM SUPERPOSITION PRINCIPLE

Statistical histogram is a very simple and useful tool in image processing which can represent the distribution of the pixel intensities in the image. In this paper, the histogram superposition principle is proposed based on the statistical characteristics of the histogram which can be described as follows: histogram is the statistical result of each pixel intensity in the image and any changes in a pixel will have an impact on the histogram. Eq.(1) describes the impact on the histogram.

$$H_{\mathrm{new}}=H\boxminus P\boxplus R \tag{1}$$

where H_{new} is the histogram after superposition, H is the original histogram, P is the histogram of the pixel set before the pixel intensities change and R is the histogram of the pixel set after the pixel intensities change. This expression can be simply interpreted as H_{new} equaling H minus ($\boxminus$) P plus ($\boxplus$) R. Note that $\boxminus$ and $\boxplus$ are two special operations. As shown in Fig.2, we call it the histogram superposition principle.

Comparing the histogram of the background image with no object and the histogram of the image that includes an object as shown in Fig.3, it is easy to find out the influence on the histogram (The parts, which the arrow points to, are the differences in histogram caused by the object) by the object. According to these differences, we can determine whether an object appeared in the scene.

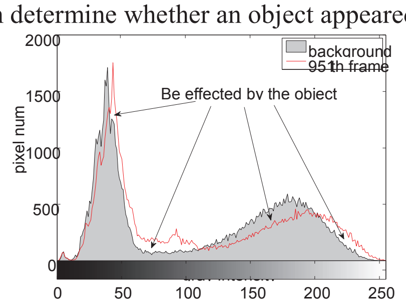


Fig.3 Gray histogram affected by the object.

IV. MULTIPLE COLOR SPACE HISTOGRAM MODEL

We propose the Multiple Color Space Histogram Model (MCSHM) based on the multiple color space and the histogram superposition principle. Robust dynamic scenes moving object detection algorithm should be insensitive to the installation position of the camera and have good adaptability to background slight changes. Building a

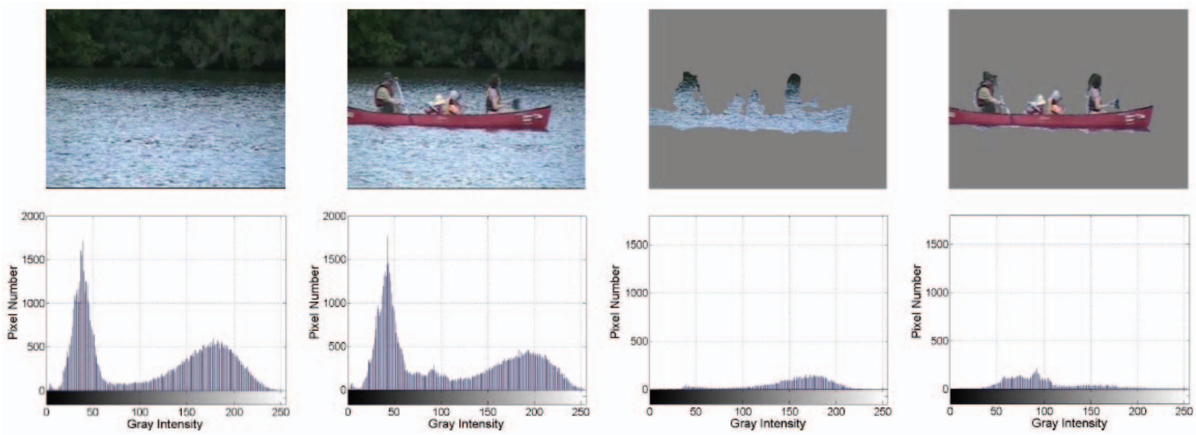


Fig.2 Video frames (first row) and gray histograms (second row), 777th frame and 951th frame of "Canoe" [23].

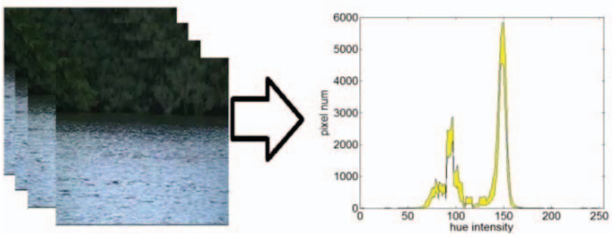
stable background model is the key to detect moving objects. MCSHM combines the stability of histogram and temporal characteristic of video sequence to construct a stable background histogram model.

A. Model initialization

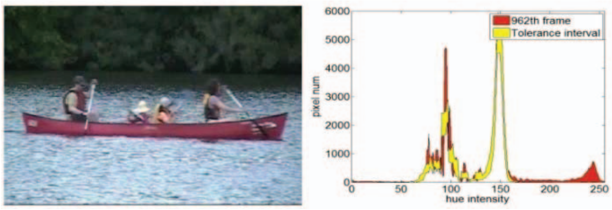
In order to obtain a stable background model we model the original MCSHM with the prior N frames of the video sequence. Firstly, mapping the frame into multiple color space Ω (in this paper, we choose a, b, H, U, and V five color components from three color spaces, Lab, HSV and YUV). Then calculate the statistical histogram of each color component. Using Eq.(2) to calculate the upper and the lower bounds of each bin in the histograms.

$$\begin{cases} u_j = \max(H_j^i) \\ l_j = \min(H_j^i) \end{cases}, i \in (0, N), j \in (0,255) \qquad (2)$$

where u and l are the upper and lower bounds of the statistical histogram of one color component in Ω respectively. H is the histogram of the color component, i is the frame index and j is the bin's index of the histogram. We call the difference between l and u as the tolerance interval. We combine the l and u as LUB and MCSHM is defined as a group of LUBs in Ω. After initialization, MCSHM tends to be stable(stable means the positions of l and u are relatively stable) as shown in the 1st row of Fig.4.



(a) Video sequences and LUB



(b) 962th frame of "canoe" and Histogram with LUB

Fig.4 The tolerance interval of hue.

In the 2nd row of Fig.4, we can see the histogram of the image exceeds the tolerance interval range, when an object appears.

B. Target detection

To detect whether an object appears in the scene. We need to compare the new histograms to the MCSHM. We use Eq.(3) to calculate the decision curve L for the histogram of each color component, which can describe the intensity of the changes.

$$L_j = \begin{cases} H_j, H_j < l_j \\ H_j, H_j > u_j, j \in (0,255) \\ 0, \ \ others \end{cases} \qquad (3)$$

where H is the histogram of the current frame. It can be seen from Eq.(3) that when $L_j \neq 0$, this indicates the changes, the number of pixels with the intensity of j, are relatively large, which implies that an object appeared in the scene (see Fig.5).

In Fig.5, the 1st row is the 831th, 871th, 981th, 1056th and 1100th frame of "canoe". The 2nd row is the decision curves of the first row, it shows that when an object appears, the decision curves have a significant fluctuation. So, according to the degree of variability of the decision line, we can determine if there is an object in the scene. To measure the intensity of curve fluctuations, a threshold is needed, which is adaptively calculated by Eq.(4).

$$T = Wimage \times Himage \times \delta \qquad (4)$$

where W_{image} and H_{image} are the width and the height of the image, respectively. δ is the minimum percentage that the number of pixels (belonging to the object) takes in the image that can be detected. Then, using Eq.(5) to determine if there is an object or not,

$$R = \begin{cases} 1, V \geq T \\ 0, others \end{cases} \qquad (5)$$

where V can be obtained through Eq.(6).

$$S = \sum_{j=0}^{255} \left| L_j \right| \qquad (6)$$

where R in Eq.(5) is the detection result, R = 1 indicates that there is an object, while R = 0 shows that there is no object. In MCSHM the number of color components determines the number of decision curves. Each decision curve can get a sub-result through Eq.(5). The final

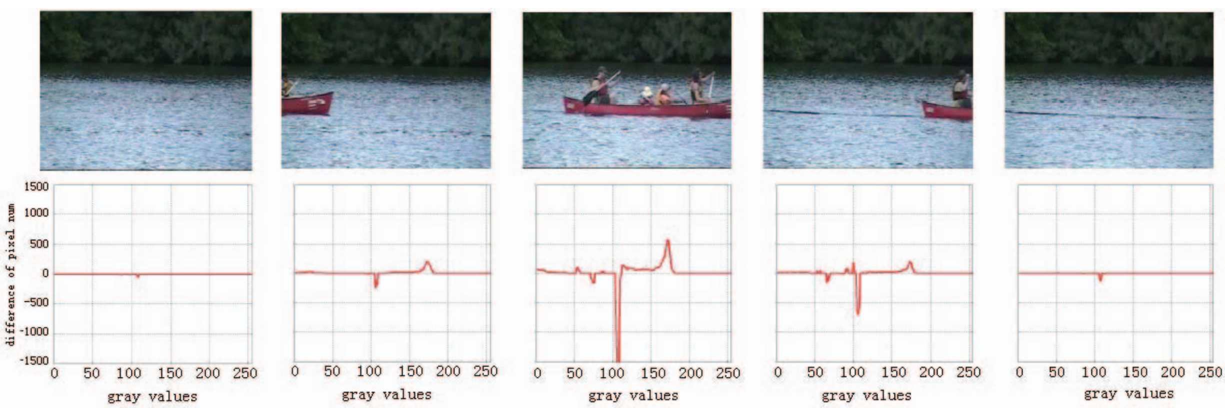


Fig.5 First Row: Video Frames; Second Row: Decision curves of Hue in HSV color space.

detection result is determined from the voting of sub-results. In Fig.6, take “canoe” as an example, the five curves in (a) are S (defined in Eq.(6)) curves of each color component in Ω. The curve in (b) is the sum of foreground pixel number of ground truth. It’s obvious that the trend of the curves both in (a) and (b) is similar. That is to say that the color components in Ω have a good reflection on the object. (c) is the voting result of S.

C. Model update

The update of the background model depends on the detection results in Section IV.*B*. MCSHM adjusts the learning rate according to the decision curves of the current frame. The update mechanism includes two aspects: One is the adjustment of the LUBs’ position and the other is the adjustment of the LUBs’ size. Here we use Eq.(7) to adjust the position of LUBs.

$$\begin{cases} u_j^{t+1} = u_j^t + \alpha L_j \\ l_j^{t+1} = l_j^t + \alpha L_j \end{cases}, j \in (0,255), \alpha \in (0,1) \quad (7)$$

where u_j^t and l_j^t are the current LUB in MCSHM, u_j^{t+1} and l_j^{t+1} are the LUB after position adjustment, L is the decision curve of the current frame and α is the learning rate(in this paper, α = 0.2). Eq.(7) can move each pair of bins in l and u upwards or downwards. This operation can make MCSHM more close to the intrinsic-MCSHM of the scene.

The adjustment of the LUB size contains two aspects, enlarge or reduce. Here, we adjust the LUB size according to the frequency of the position adjustment of LUB. When the frequency of the position adjustment is very high, we consider that the current size of LUB cannot adapt to the current background. So, the size of LUB needs to be expanded to adapt to the background. Instead, if the position of the LUB has never been adjusted within a time window, it is valid to consider that the LUB size is too large to accurately describe the background. Therefore, it needs to be appropriately reduced. Eq.(8) is used to adjust the size of LUB.

$$\begin{cases} u_j^{t+1} = u_j^t + S \\ l_j^{t+1} = l_j^t - S \end{cases} \quad j \in (0,255) \quad (8)$$

where S is the step-size calculated by Eq.(9).

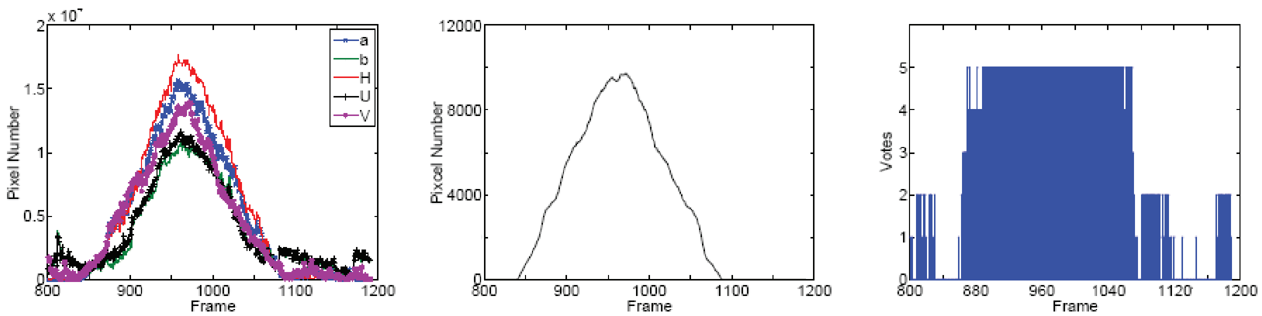
$$S = \begin{cases} \beta(u_j^t - l_j^t)e^{q_j - 1}, & q_j > \varepsilon \\ 0, & 0 < q_j \le \varepsilon \\ -\beta(u_j^t - l_j^t)e^{p-1}, & q_j = 0, p < \varepsilon \end{cases} \quad j \in (0,255), p \in (0,1) \quad (9)$$

where q is the frequency of location adjustment of LUB, p a random number from 0 to 1 and ε is a threshold that is taken as 0.03 in this paper.

V. EXPERIMENTS AND RESULTS

In this paper, we propose a novel algorithm to detect moving objects in dynamic scenes. As compared to the other algorithms, our algorithm focuses on whether there is a moving object in the scene or not and do not care about the location of the object. In fact, in the application of video surveillance, this idea is more in line with the needs of people. Because the detection mechanism of our algorithm is different from that of other algorithms, the output form is different (our algorithm’s output is a Boolean value, true or false whereas others’ are the binary images), too. Therefore, we need to unify the output form. In this paper, we unify the output form into the Boolean value. So we add a parameter, t(the percentage that the foreground-pixel number takes in the image), with each algorithm. If t is greater than the predetermined threshold value, it means that there is an object. Otherwise, no. In order to evaluate the performance of our algorithm with other algorithms fairly, we set the predetermined threshold value in other algorithms to the same value as the threshold value δ of our algorithm.

We use the false positive (FP), false negative (FN), true positive (TP), true negative (TN), precision, recall and F-measures to analyze the performance of the algorithms.
Where the true means the correct classification, false is the false classification, positive indicates there is an object in the scene and negative indicates that there is no object in the scene. We compared our method with five other methods on ChangeDetection.Net(CDNET) [23]. Table I shows the arguments setting of each algorithm. We use three typical scenarios to compare the performance of the algorithms.



(a) (b) (c)

Fig.6 Detection result of MCSHM. (a) S curves of each color component. (b) S curve of ground truth. (c) Voting result of S.

TABLE I. ARGUMENTS SETTING OF EACH ALGORITHM.

Method ID	Settings
DPZivkovicAGMM [5]	$T = 20, \alpha = 0.01, n = 3$
FuzzyChoquetIntegral [24]	$T = 0.67, LF = 10, \alpha_{learn} = 0.5, \alpha_{update} = 0.05$, RGB+LBP
MOGV2 [5]	$T = 5, \alpha = 0.01$
SJN-MultiCue [25]	Default parameters from [23]
PBAS [19]	Default parameters from [24]
MCSHM	$\alpha = 0.2, \varepsilon = 0.03, \Omega =$ {a, b, H, U, V} , (**Lab**+**HSV**+Y**UV**)

A. Leaves swaying

We use "overpass" [23] to test the algorithms' adaptability to the scene of swaying leaves. In Fig.(7), DPZivkovic-AGMM and MOGV2 are the improvement algorithms of GMM. Therefore, both of them have the same problem that there are holes inside the target. This issue is caused by the update mechanism of this kind of method. The background recovery of FuzzyChoquetIntegral is too slow so there are some holes left in the background after the man left in 2699th frame. Both SJN-MultiCue and PBAS are robust to this type of dynamic scene. But in 2507th frame, PBAS lose too much foreground pixels. To give a qualitative understanding of the performance of MCSHM, we just showed the decision curve of the hue component of **Ω**. It can be seen that when there is no target in the scene, the curve is very smooth and steady. When the target appears, a violent fluctuation arises in the curve. As the object fades away, the curve is synchronized to reinstate the previous state. It shows that our method can accurately describe the influence of the object on the background and can determine whether there is an object in the scene or not.

Table II presents the performance metrics for the algorithms applied to "overpass". MOGV2 and PBAS missed too many foreground pixels that lead to very low accuracy rate. FuzzyChoquetIntegra's low accuracy rate results from its low background recovery rate. Table II shows that our method outperforms the other methods for precision and FPS.

TABLE II. EVALUATION RESULTS FOR "OVERPASS" OF EACH ALGORITHM ($\delta = 0.06$).

Method ID	TP	TN	FP	FN	Precision	Recall	F-measures	FPS
DPZivkovicAGMM	367	1623	2	9	0.9946	0.9761	0.9852	178.6
FuzzyChoquetIntegral	**376**	1438	187	**0**	0.6679	**1.0000**	0.8009	26.7
MOGV2	93	1624	1	283	0.9894	0.2473	0.3957	219.8
SJN-MultiCue	375	1605	20	1	0.9494	0.9973	0.9728	34.9
PBAS	150	1624	1	226	0.9934	0.3989	0.5693	64.7
MCSHM	366	**1625**	**0**	10	**1.0000**	0.9734	**0.9865**	**287.7**

B. Waves rippling

"Canoe" is used to test the algorithms' adaptability to waves rippling situation. **Fig.8 shows** DPZivkovicAGMM and FuzzyChoquetIntegral are well adaptable to this type of scene. Our method can also handle this kind of a scene very well.

In Table III, it can be seen that our method has a very close performance with that of DPZivkovicAGMM in accuracy rate, recall rate and F-measures. But our method outperforms all other methods for FPS.

TABLE III. EVALUATION RESULTS FOR "CANOE" OF EACH ALGORITHM ($\delta = 0.02$).

Method ID	TP	TN	FP	FN	Precision	Recall	F-measures	FPS
DPZivkovicAGMM	**204**	180	6	0	0.9714	**1.0000**	**0.9855**	181.8
FuzzyChoquetIntegral	197	**186**	**0**	7	**1.0000**	0.9657	0.9825	28.1
MOGV2	111	**186**	**0**	93	**1.0000**	0.5441	0.7048	185.2
SJN-MultiCue	42	**186**	**0**	162	**1.0000**	0.2059	0.3415	29.8
PBAS	111	**186**	**0**	93	**1.0000**	0.5441	0.7048	63.5
MCSHM	201	181	5	3	0.9757	0.9853	0.9805	**291.5**

TABLE IV. EVALUATION RESULTS FOR "OFFICE" OF EACH ALGORITHM ($\delta = 0.02$).

Method ID	TP	TN	FP	FN	Precision	Recall	F-measures	FPS
DPZivkovicAGMM	1162	39	3	277	0.9974	0.8075	0.8925	188.7
FuzzyChoquetIntegral	1438	22	20	1	0.9863	0.9993	0.9928	21.8
MOGV2	122	**42**	**0**	1317	**1.0000**	0.0848	0.1563	235.3
SJN-MultiCue	**1439**	33	9	**0**	0.9938	**1.0000**	0.9969	54.9
PBAS	789	32	10	650	0.9875	0.5483	0.7051	62.0
MCSH	1437	40	2	2	0.9986	0.9986	**0.9986**	**294.5**

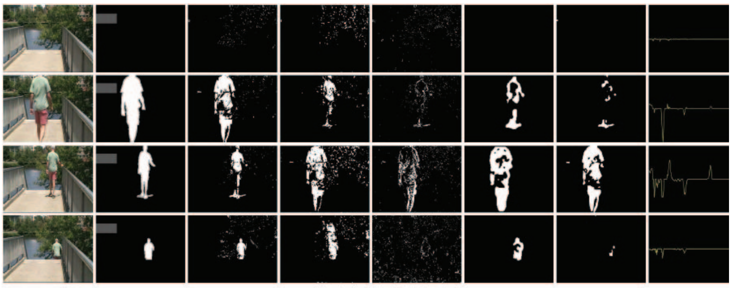


Fig.7 Sample outputs for the video sequence of "overpass". First Column: Input Frames (1703th, 2398th, 2507th, 2699th); Second Column: Ground Truth; Third Column: DPZivkovicAGMM; Fourth Column: FuzzyChoquetIntegral; Fifth Column: MOGV2; Sixth Column: SJN-MultiCue; Seventh Column: PBAS; Eight Column: MCSHM (Decision Line of Hue).

C. Static scene

We test the algorithms' adaptability to static scenes on "office"[23]. Fig.9 shows MOGV2, FuzzyChoquet-Integral and PBAS cannot deal with static background well whereas our method has a good adaptability to the static scenes.

MCSHM has very good adaptability to static scenes because of the good stability of the static background. It can be seen in Table IV that MCSHM shows the best performance.

VI. CONCLUSION

In this paper, we proposed a novel algorithm for detecting whether a moving object appeared in dynamic scenes. The algorithm models the background through the adoption of multiple color space and histogram superposition principle. The algorithm can adapt the learning rate dynamically, by which the background model is updated frame-by-frame. Our method models the background based on histogram which is different from the pixel-level modeling method. Therefore, the computational complexity of the proposed algorithm is very low. Thanks to the flexibility of the multiple color space, this algorithm is also very flexible. Choosing appropriate color components can enhance the robustness of our method. But there are still some

Fig. 8. Sample outputs for the video sequence of "canoe". First Column: Input Frames (837th, 902th, 964th, 1048th); Second Column: Ground Truth; Third Column:

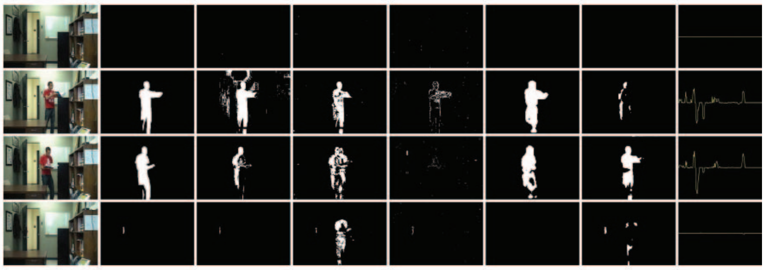


Fig.9 Sample outputs for the video sequence of "office". First Column: Input Frames (571th, 771th, 1360th, 2042th); Second Column: Ground Truth; Third Column: DPZivkovicAGMM; Fourth Column: FuzzyChoquetIntegral; Fifth Column: MOGV2; Sixth Column: SJN-MultiCue; Seventh Column: PBAS; Eight Column: MCSHM (Decision Line of Hue).

problems with this method, for instance, how to select the appropriate color components for different scenes, how to weigh the contribution of different color components, and how to determine the value of the threshold? Further research will be carried out on these three issues in the future.

REFERENCES

1. Ridder, C., Munkelt, O., Kirchner, H. Adaptive background estimation and foreground detection using Kalman-filtering. In: Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on. (1995) 390–395
2. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P. Pfinder: real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 780–785
3. Stauffer, C., Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Volume 2. (1999) 246–252
4. Stauffer, C., Grimson, W.E.L. Learning patterns of activity using real-time tracking. Pattern Analysis and Machine Intelligence IEEE Transactions on **22** (2000) 747–757
5. Zivkovic, Z., Heijden, F.V.D. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27** (2006) 773–780
6. Chan, A.B., Mahadevan, V., Vasconcelos, N. Generalized Stauffer-Grimson background subtraction for dynamic scenes. Machine Vision and Applications **22** (2011) 751–766
7. Wang, S.C., Su, T.F., Lai, S.H. Detecting moving objects from dynamic background with shadow removal. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (2011) 925–928
8. Zhao, Z., Bouwmans, T., Zhang, X., Fang, Y. A fuzzy background modeling approach for motion detection in dynamic backgrounds. Communications in Computer and Information Science **346** (2012) 177–185
9. Liu, G., You, Y., Zheng, S., Li, F. Moving Target Detection Based on Improved Mixture Gauss Model. Springer Berlin Heidelberg (2012)
10. Chen, Z., Ellis, T. A self-adaptive Gaussian mixture model. Computer Vision and Image Understanding **122** (2014) 35–46
11. Varadarajan, S., Wang, H., Miller, P., Zhou, H. Fast convergence of regularized region-based mixture of Gaussians for dynamic background modeling. Computer Vision and Image Understanding **136** (2015) 45–58
12. Shoushtarian, B., Bez, H.E. A practical adaptive approach for dynamic background subtraction using an invariant color model and object tracking. Pattern Recognition Letters **26** (2005) 5–26
13. Zhang, S., Yao, H., Liu, S., Chen, X., Gao, W. A covariance-based method for dynamic background subtraction. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. (2008) 1–4
14. Zhang, S., Yao, H., Liu, S. Dynamic background subtraction based on local dependency histogram. International Journal of Pattern Recognition and Artificial Intelligence **23** (2011) 1397–1419
15. Barnich, O., Droogenbroeck, M.V.: Vibe. A powerful random technique to estimate the background in video sequences. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. (2009) 945–948
16. Barnich, O., Van, D.M. Vibe: a universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society **20** (2011) 1709–1724
17. Van Droogenbroeck, M., Paquot, O. Background subtraction: Experiments and improvements for Vibe. Computer Vision & Pattern Recognition Workshops , 2012 , 71 (8) :32-37
18. Chen, X., Liao, J., Li, B., Chen, Q. Foreground detection based on modified Vibe in dynamic background. Optics and Precision Engineering **22** (2014) 2545–2552
19. 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. (2012) 38–43
20. Wang, H., Suter, D. Background subtraction based on a robust consensus method. In: 18th International Conference on Pattern Recognition (ICPR'06). Volume 1. (2006) 223–226
21. Yoshinaga, S., Shimada, A., Nagahara, H., Taniguchi, R.I. Object detection based on spatiotemporal background models. Computer Vision and Image Understanding **122** (2014) 84–91
22. Sobral, A., Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Computer Vision and Image Understanding **122** (2014) 4–21
23. Pierre-Marc, J., Prakash, I., Janusz, K., Porikli, F. Changedetection.net(cdnet). Website (2014) http://wordpress-jodoin.dmi.usherb.ca/dataset2014.
24. Baf, F.E., Bouwmans, T., Vachon, B. Fuzzy integral for moving object detection. In: Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on. (2008) 1729–1736
25. Noh, S., Jeon, M. A new framework for background subtraction using multiple cues. In: Asian Conference on Computer Vision. (2012) 493–506