Robust moving object detection based on ViBe with adaptive shadow detector

ZHIHUI FAN*, ZHAOYANG LU, JING LI, CHAO YAO, WEI JIANG
State Key Lab.of Integrated Service Networks
Xidian University
No.2 South Taibai Road, Xi'an,Shaanxi 710071
CHINA
* Corresponding author, e-mail: fanzhihui0379@126.com

Abstract: - We propose the SAViBe+ algorithm, a new approach for moving object detection based on ViBe background subtraction algorithm with an adaptive shadow detector. Because ViBe cannot handle scenes containing gradual illumination variations, and eliminate shadows cast by moving objects, an adaptive shadow detector is designed to detect and eliminate the shadow of a moving object, adapting to variation of illumination in an automatic manner, which adopts texture and spatiotemporal information. This adaptive shadow detector is built with a texture model (TM) and a hue model (HM) to estimate the texture and intensity change of false foreground pixels respectively. A factor called Mean of Value (MofV) is proposed to work with HM to improve its efficiency. This algorithm is robust against false detection for different types of videos in indoor and outdoor scenes under various types of illumination taken by stationary cameras. Quantitative and qualitative performance evaluation carried out on the database of Change Detection Workshop(CDW'14) revealed that our scheme could operate in real-time, rapidly adapt to variation of illumination and environment online and outperform state-of-the-art methods.

Key-Words: - Object detection, Shadow removal, Adaptive shadow detector, Background subtraction, Motion detection, Surveillance

1 Introduction

Detection of moving objects in video is an important procedure for computer vision applications [1], such as video surveillance, human-machine interface and event detection. Among approaches of detecting moving objects, background subtraction is one of the most popular methods in practical applications for its efficiency and simplicity, which compare a static background frame with the current frame of a video scene, pixel by pixel to detect a pixel belonging to the background or to the foreground. Detection of moving objects in video is also referred to motion detection. Background changes as shadows make negative effects to it. Since the shadow cast by a moving object is mostly connected to the object and significantly differs from the background, it is frequently misclassified as part of foreground objects, which makes it difficult to detect the exact shape of objects and recognize the objects. Therefore, the accurate detection of a moving object and the acquisition of its exact shape by eliminating background changes bring a positive effect on the performance of subsequent steps such as tracking, recognition, and activity analysis.

Generally speaking, background changes that motion detection algorithm faces can be divided into two types [2]: illumination changes-changes relating to lighting conditions such as sun rising or setting, being blocked by clouds; dynamic changes-changes relating to the swaying motion of tree branches, leaves and grass, waves water. Several methods have been proposed for shadow removing[3-6]. Recent surveys [7–10] analyze the variety and range of alternative approaches. To date, no motion detection algorithm is robust under the all above conditions.

Recently, a background modeling method called ViBe (Visual background extractor) based on probability and statistics has attracted more and more attention due to its extremely high speed. ViBe algorithm stores, for each pixel, a set of values taken in the past at the same location or in the neighborhood. It then compares this set to the current pixel value in order to determine whether that pixel belongs to the background, and adapts the model by choosing randomly which values to substitute from the background model[11]. The algorithm can suppress environment dynamic changes described above and the impact of camera jitter on foreground detection effectively with better performance compared with other methods. Nevertheless, the performance of ViBe is poor when applied to complex background, shadow scenes.

Several related work is proposed on this issue. Li et al. [12] proposed an improved ViBe algorithm based on adjacent frame difference algorithm, which can quickly remove the ghost region and improve the detection accuracy . M. Van Droogenbroecket al.[13] introduced several modifications to ViBe, such as distinction between segmentation mask and updating mask, increased adapted distance measure and an increased updating factor, to enhance the algorithm adaptability. QIN et al. [14] described an improved ViBe algorithm in HSV color space, attempting to eliminate shadows of the foreground objects, which employed the ratio of V, difference of H and S between foreground and background to determine whether the interest pixels detected by ViBe are shadows. The results indicated no satisfactory performance when confronting with shadow scenes, thus Vibe needs to be further improved.

Nowadays background modeling methods integrating multiple image-level information have drawn considerable attention due to their effectivity and efficiency, relying on exploring advantages from more than one image level[15-20]. As the texture of the shadow is stationary related to the background compared to color[21], especially in an outdoor environment, spatial information can absorb the effects of illumination changes[17], and temporal information is useful in handling the dynamic changes[1]. So a novel robust algorithm for moving object detection and shadow removal algorithm adopting texture and spatiotemporal information is proposed in this article. This new algorithm is called SAViBe+ (Enhanced Self-Adaptive ViBe algorithm for moving object detection) .

This paper is organized as follows. In Section 2, we describe the shadow illumination model. Chapter 3.1 describes system framework of the proposed method. Chaps 3.2 and 3.3 respectively discusses the proposed texture model (TM) and hue model (HM). We provide a thorough analysis of experimental results on the CDW’14 dataset and comparisons with several state-of-the-art methods in Section 4 . Section 5 draws the conclusion.

2 Shadow illumination model

Most shadows encountered in the real world is shown in Fig.1a and Fig.1b. Fig. 1a presents the outdoor scenes. Shadows of objects are mainly produced only by sunlight which is usually assumed as parallel light. Fig.1b shows the indoor scenes or nighttime outdoor scenes, such as hall or streets at night. Shadows of objects are produced by more than one light source. These illumination models have following assumptions:(1)The camera is fixed. (2)The background and moving object have Lambertian surfaces.(3) Every object in the scene is perfectly opaque and therefore does not transmit the light.

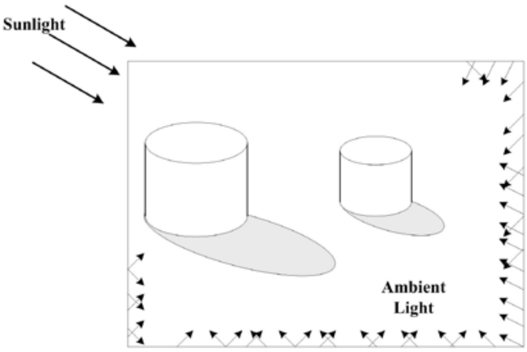


Fig.1a. Illumination model (Sunlight only).

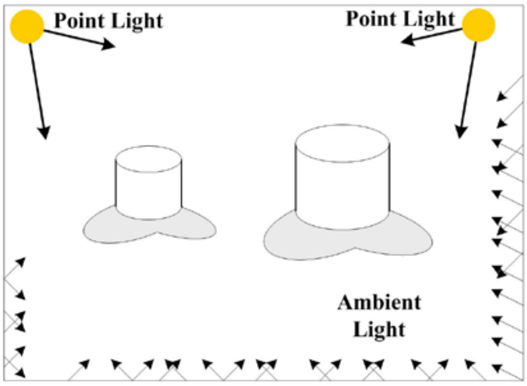


Fig.1b. Illumination model (multi-light sources).

There are two kinds of illumination in the shadow model Fig.1a: one is a single visible light point source from the sun (sunlight),and the other is a diffusion extended light source from the sky (ambient light). Every point in the scene has the same ambient light. The rays of sunlight are mutually parallel because they are from a far distant point source. Thus, every pixel of the bright object and bright background has the same sunlight. There exists more than two kinds of illumination in the shadow model Fig.1b:at least two of the light sources are point lights and the other is a diffusion extended light source from the surrounding (ambient light). Every point in the scene has the same ambient light. The rays of point-light are not parallel.

Shadow models to be dealt with in this article is shown in Fig.1. Obviously, a shadow is a region of relative darkness that occurs when an object totally or partially occludes direct light from light sources.The shadow generated by an object may be classified as either ‘self’ or ‘cast’; self shadow occurs on the object occluding the light. As self shadow is of less importance for discriminating false foreground pixels, here we do not discuss them .

3 The SAViBe+ algorithm

3.1 System Framework

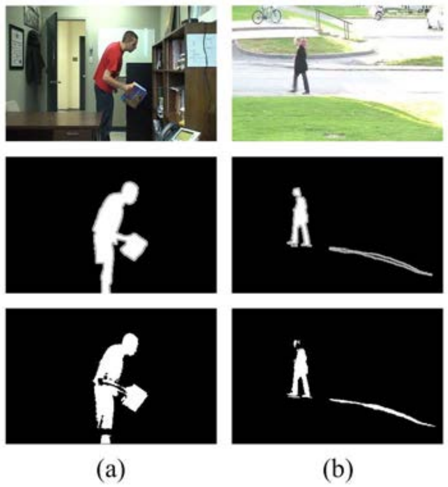


Fig.2. Performance of ViBe (①top row is the input, middle row is the groundtruth, and bottom row is the output;②white points are foreground, and gray points are shadows.).

As to ViBe, during the detection of moving objects, if illumination does not change, only moving object pixels are detected as foreground pixels (see Fig. 2a). However, if illumination changes, shadow pixels would also be judged as foreground pixels (see Fig. 2b), which will confuse the action analysis and event detection. These false foreground pixels should be eliminated. To discriminate actual moving object pixels and their casting shadows, we develop TM , HM and *MofV*. Fig.3 indicates a flow diagram of our scheme: ViBe combined with TM cascading by HM with a factor called Mean of Value(*MofV*) in HSV color space. For frame sequence, first, we utilize ViBe and TM to detect foreground simultaneously, and integrate their detection results by following operations. In the segmentation silhouettes or bounding boxes of objects produced by TM, AND operations are implemented to obtain sub-objects, and these sub-objects are combined by location. Through this step, almost all the interior shadows shown in Fig.1b can be eliminated. For the shadows resulted in Fig.1a, HM is further employed in our scheme. The factor *MofV* in HSV is adopted to determine which pixels should be further determined by HM.

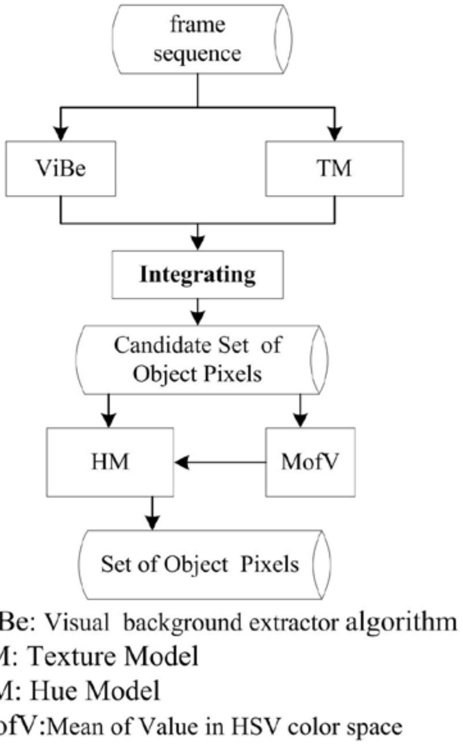

frame sequence
ViBe
TM
Integrating
Candidate Set of Object Pixels
HM
MofV
Set of Object Pixels
ViBe: Visual background extractor algorithm
TM: Texture Model
HM: Hue Model
MofV:Mean of Value in HSV color space

Fig.3. Scheme of the proposed method.

Texture model TM and Hue model HM will be detailed in Chapter 3.2 and Chapter 3.3 respectively.

3.2 Texture model

According to the reflection and shadow illumination model[22] which forms the basis of many shadow detection algorithms, generally we assume shadows do not change the texture of images. Here the spatial information is utilized to adapt to illumination changes. In [20], Liao devised a texture operator called SILTP, a scale invariant local ternary pattern, and shows its effectiveness for handling illumination variations. In this section, a detail depiction of TM is given based on SILTP.

3.2.1 SILTP operator

Illumination variations, either global or local, often cause sudden changes of gray scale intensities of neighboring pixels simultaneously, which would approximately be a scale transform with a constant factor. The intensity scale invariant property of a local comparison operator is quite important. Given any pixel location (x_c, y_c), SILTP encodes it as (1).

$$SILTP^{\tau}_{\mathrm{N,R}}(\mathrm{x_c},\mathrm{y_c}) = \bigoplus_{\mathrm{k=0}}^{\mathrm{N-1}} s_\tau(I_C, I_K), \qquad (1)$$

where I_c is the gray intensity value of the center pixel, I_K notates one of its N neighbors equally spaced on a circle of radius *R*, ⊕ denotes concatenation operator of binary strings, τ is a scale factor indicating the comparing range, and S_τ is a piecewise function defined as:

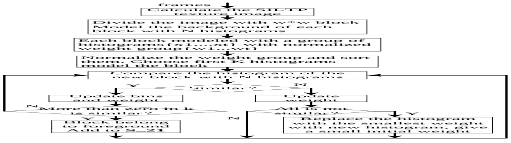
$$S_\tau(I_C, I_K) = \begin{cases} 10, & if\ I_K < (1+\tau) I_{C,} \\ 01, & if\ I_K > (1+\tau) I_{C,} \\ 00, & otherwise. \end{cases} \tag{2}$$

The scale invariance of SILTP operator can be easily verified[20]. The SILTP feature is invariant if the illumination is suddenly changed.

3.2.2 Texture model

To quickly segment the foreground and background, we model the texture background in a block-wise manner which accelerates the running speed. We divides each new video frame into blocks with equal size by a partial overlapping grid structure. The feature vector of a particular image block over time is considered as a block process. Since we refer SILTP histograms as the feature vector, the block process is defined as a time series of SILTP histograms. We denote the block histogram at time instant *t* by x_t. Each block is modeled by a group of *K* weighted SILTP histograms $\{x_1, ..., x_t\}$. We appoint the weight of the k^{th} histogram at time instant *t* by $\omega_{k,t}$. The initial value of $\omega_{k,t}$ is generated randomly, and all the $\omega_{k,t}$ of each block normalized to 1. We model and update background for each block in the way like Heikkilä M et al in [23].

To discriminate one block in a new frame belonging to moving object or background, the first thing is to compare the new block histogram x_t against the existing *K* model histograms by a distance measure. The histogram correlation is employed as the distance measure in our experiments in which the normalized histograms h_1 and h_2 is specified as Eq. (3), where *i* is the bin index of the histogram, *B* is the max number of bin.


frames
Calculate the SILTP texture image
Divide the image with w*w block Model the background of each block with N histograms
Each block modeled with a group of histograms{x1,...xt} with normalized weight group{w1,...wt}
Normalize the weight group and sort them, Choose first K histograms model the block
Compare the histogram of the new block with N histograms
Similar?
Y
N
Update bins and weight
Update weight
N
More than zero in k is similar?
Y
All is not similar?
Y
N
Block belong to foreground Add to S_21
Replace the histogram with the smallest weight with new histogram, give a small initial weight

(3)

If none of the model histograms is close enough to the new histogram, the model histogram with the lowest weight is replaced by the new histogram and given a low initial weight. Afterward, the weights are normalized to confirm their sum equal to one. If a model histogram close enough to the new histogram is found, the bins of this histogram are updated. The histograms are sorted in decreasing order according to their weights. As a result, the most probable background histograms are on the top of the list. As a last phase of the updating procedure, the first *B* histograms are selected to be the background model. Foreground detection is achieved via comparison of the new block histogram x_t against the existing *B* background histograms selected at the previous time instant. If none match is found, the block is considered to belong to the foreground. Otherwise, the block is marked as background. TM obtains the foreground by the above process.

By integrating the detection results of ViBe and TM as the way described in Chap.3.1, foreground can be detected exactly when facing illumination in Fig.1b. However, there may exist some shadows pixels in the foreground when facing illumination in Fig.1a. In Chapter 3.3 we will introduce the development of the HM to distinguish other remaining false foreground pixels.

3.3 Hue model

The proposed shadow removal method is developed according to the following two facts. The brightness of the shadow region is lower than that of the background area. Moreover, chromaticity is the essential attribute of material and keeps constant with respect to illumination changes. However, environmental disturbance makes it a little floating, so we employ Gaussian model on chrominance of frame sequence in pixel-wise way. For Chrominance and luminance information can be effectively separated in the HSV color space, HSV color space is commonly used to detect and remove shadows.

The shadow pixels are always darker than the background ones. Therefore, a pixel which becomes brighter cannot be a shadow pixel. So we consider a pixel in the set of moving pixels which is brighter than the background as a moving object pixel. If a pixel is darker than background, it may be moving objects with dark material or shadows. To discriminate these pixels in foreground pixels, we develop HM for each pixel which is modeled with single Gaussian.

3.3.1 Hue model

In the following, we explain the HM updating procedure for one pixel, but the procedure is identical for each pixel. We make use of first *N* frames to estimate the mean $m_H(x,y)$ and the standard deviation $\sigma_H(x,y)$. Then they are updated as:

$$m_{H,t} = (1 - H_\alpha) m_{H,t-1} + H_t(x,y), \qquad (4)$$

$$\sqrt{\sigma_{H,t}} = (1 - H_\alpha)\sqrt{\sigma_{H,t-1}} + \left(H_t(x,y) - m_{H,t}\right)^2, \quad (5)$$

where H_α is the learning rate which is set by the user.The smaller the value, the more conservative the algorithm updates. $H(x,y)$ denotes the hue value of each pixel p(x,y). Using the estimated $m_H(x,y)$ and $\sigma_H(x,y)$, we calculate the threshold T_h^K and T_l^K as:

$$T_h = \hat{m}_H + 2.58 * \hat{\sigma}_H, \qquad (6)$$

$$T_l = \hat{m}_H - 2.58 * \hat{\sigma}_H, \qquad (7)$$

with reliability of 99.7%; P(-2.58 < *Z* < 2.58) = 0.9973, *Z* = *N*(0,1). With T_h and T_l, by verifying whether or not the $H(x,y)$ of a pixel is in the feasible range, the pixel is further discriminated by

$$\begin{cases} p(x,y) \in S, & if\ T_l < H\ (x,y) < T_h, \\ p(x,y) \in O, & otherwise. \end{cases} \qquad (8)$$

where S is the set of shadow pixels and O the object pixels.

Hue model is used to determine a pixel p(x,y) be a moving object pixel or a shadow pixel. To improve efficiency, all of the pixels in the candidate set of foreground pixels produced by ViBe and TM should not be further discriminated by HM, thus we need to find which pixels are probable shadow pixels. The factor *MofV* in HSV is exploited.

3.3.2 The factor MofV

The factor *MofV* is employed to determine which pixels to be further detected by HM. The spatiotemporal information is united employed here. Region-level information is used as a supplement to pixel-level statistical information, adapting to illumination changes which are not observed in previous frames. A preprocessing step, we divide the image into several pixel clusters. Koppal et al in[24] divided a complex scene into geometrically consistent clusters (scene points that have the same or very similar surface normals) irrespective of their material properties and lighting. The results of [24] show that the pixels which are positionally and chromatically close to each other belong to the same cluster. Here, we employ K-means clustering to acquire the clusters utilizing a chromatic and positional feature. The number of clusters depends on the size of image, we define the size of cluster *Rect*=40*40. In the following, we introduce how to initialize and update the *MofV*. The values of its past and its cluster are utilized to tolerate the effects of both illumination and dynamic changes. The *MofV* of each pixel p(x,y) is expressed as $MofV(x,y)$, and it is defined as follows:

$$MofV(x,y) = \frac{1}{N+M}\left(\sum_{i=1}^{N} V_i + \sum_{q \in C} V_q\right), \qquad (9)$$

where *C* is a pixel cluster, *M* is the number of pixels belonging to *C*, *N* is the past frames, V_q and V_i are the value of pixel q and pixel i in HSV respectively.

For a new frame, we define $d(x,y)$ as the distance of the value of a new pixel to its *MofV*, and gain $d(x,y)$ by Eq.(4) :

$$d(x,y) = MofV(x,y) - V_{p_{new}(x,y)}, \qquad (10)$$

where $V_{P_{new}(x,y)}$ is the value of the new pixels $p_{new}(x,y)$.

We use the user-defined threshold T_{Vd} to make a decision that $p_{new}(x,y)$ need to be further detected by HM($|d(x,y)| \geq T_{Vd}$) or not($|d(x,y)| < T_{Vd}$). Here, we use $|d(x,y)|$ instead of $d(x,y)$ to deal with highlight pixels simultaneously, even though most highlight pixels have been eliminated by TM.

$MofV(x,y)$ is update as Eq.(11).

$$MofV_t(x,y) = \begin{cases} (1 - V_\alpha) MofV_{t-1}(x,y) \\ + V_\alpha \cdot V_{P_{new}(x,y)} \quad if\ |d(x,y)| \geq T_{Vd}, \\ MofV_t(x,y), \qquad otherwise. \end{cases} \qquad (11)$$

By the HM which processes this execution for every frame the rest shadows can be separated from moving object pixels successfully.

4 Experimental results

The performance of the proposed method is evaluated using eight video sequences containing complex backgrounds, which are publicly available made by Change Detection Workshop [25]. The test databases are difficult video including indoor and outdoor scenes, moving cast shadows, illumination changes, camera jitter etc.

The proposed approach is compared with other three state-of-the-art background subtraction algorithms, including mixture of Gaussian (“MoG”) [22], Visual background extractor (“ViBe”) [11] and Pattern Kernel Density Estimation Local (“PKDE”) [20] . All tested algorithms are implemented in VC6.0 and opencv1.0 and run on a standard PC with 2.8 GHz, 1GB memory, and windows XP operating system. Morphological operation was not employed in the experiment, and a standard OpenCV postprocessing was used which eliminated small pieces less than 15 pixels.

Parameters for MoG, ViBe, and PKDE are all listed in Tab.1, which were suggested by the authors in their experiments. For the proposed method SAViBe+, we use a set of consistent parameters for all experiments listing in the bottom row of Tab.1. The parameters were set empirically in order to balance the trade-off between efficiency and accuracy. The background subtraction results were compared both in subjective and objective ways. Chapter 4.1 depicts the qualitative comparison results. Chapter 4.2 shows the objective evaluation results.

Table 1. Parameters for algorithms

Method	Initial Frames	Size of block	Parameters
MoG	200	Single Pixel	$win\_size = 200, n\_gauss = 3, bg\_threshold = 0.7$ $td\_threshold = 3.5, \min Area = 15$ $weight\_init = 0.05, \mathrm{var}iance\_init = 30$
ViBe	1	Single Pixel	$N = 20, R = 20, \theta = 16, \lambda = 2$
PKDE	4	3*3	$k = 3, T_b = 0.7, T_s = 0.01, T_m = 0.01, \alpha = 0.005$ $a = 0.36, b = 0.08$
SAViBe+	100	Single Pixel & 12*12	Parameters for ViBe : $N = 20, R = 20, \theta = 16, \lambda = 2$ Parameters for TM： $T_D = 0.9, T_B = 0.5, V_b = 0.005, V_w = 0.005, N = 4$ $SILTP_{P,R}$:$\mathrm{P} = 4, \mathrm{R} = 1, \tau = 0.1$ Parameters for HM： $H_\alpha = 0.05, H_\lambda = 2.58, T_{Vd} = 0.55, V_\alpha = 0.01$

4.1 Qualitative comparison

The foreground segmentation results of the compared algorithms are demonstrated with one frame for each dataset in Fig. 4.

The outdoor scene “backdoor” shown in Fig.4 contains illumination changes. As demonstrated, MoG and ViBe cannot appropriately adapt to the illumination changes and discriminate the object from background. PKDE and the proposed method SAViBe+ easily absorb shadows. However PKDE is unable to suppress the waving branches and also loses some detail information of objects.

For the outdoor scene “busStation”, there is deep cast shadows caused by hard sunlight, MoG performs worst, and ViBe detects the cast shadow of the man as another person. PKDE cannot absorb shadows completely. SAViBe+ easily eliminate all the cast shadows.

The outdoor scenario “pedestrians” is shown in Fig.4, containing moving shadows. Obviously, MoG and ViBe fail to remove the shadow of the object. In contrast, PKDE and SAViBe+ could perfectly eliminate the moving shadows. Compared with groundtruth images, SAViBe+ obtains more abundant information than PKDE, excellently modeling the background and generating accurate results.

As for the case of outdoor scene “WinterDiverway” in Fig.4, it describes a challenging dataset for background modeling. MoG and ViBe are still affected by illumination changes, and PKDE can successfully remove the shadows with nearly losing the contour of the car. Whereas, SAViBe+ can separate the moving car from the background almost perfectly.

The indoor busy scene “PETS2006” shown in Fig.4 contains moving shadows. There is more than one light sources as shown in Fig.1b. As illustrated, MoG misclasifies shadows as part of the moving object. ViBe fails to suppress the moving shadow as well. PKDE and SAViBe+ perfectly removes the shadows. However PKDE losses the inside information.

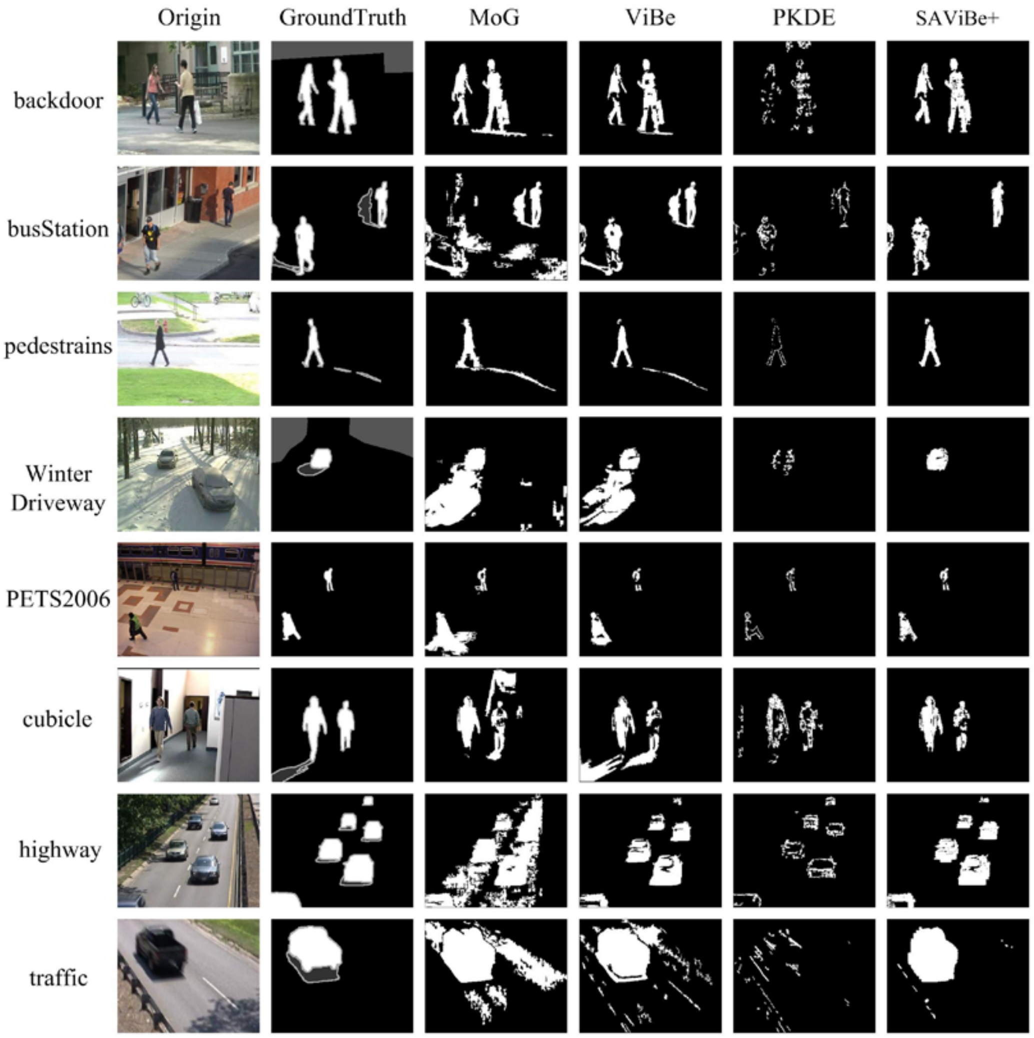


Fig.4. Performance for each method on eight datasets.

A frame of another indoor scene “cubicle” dataset shown in Fig.4 indicates complicated illumination changes caused by light changing and people walking. As can be seen, MoG cannot adapt to the illumination changing, ViBe misclassifies cast shadows as part of object as before, and PKDE has no capacity to remove the illumination changes at the bord of the door. SAViBe+ still has a better performance in removing the shadows although losing little details of objects.

Fig.4 demonstrates the common frame of “highway” dataset, which has little moving shadow and camera jitters because of wind. As illustrated, all these methods could conquer camera jitter and branches swinging except MoG and obtain accurate segmentation results. Illumination changes caused by swinging branches and camera jitter exercise a great influence on MoG, where no object is detected but false foreground.

As for the case of outdoor dynamic scene, the last row of Fig.4 shows a dataset containing a fierce camera jitter, a quite difficult dataset for background modeling. As demonstrated, PKDE and ViBe perform unsatisfactorily. MoG successfully learns the camera jitter, but cannot adapt to the illumination changes caused by the camera jitter, thus it could not obtain expected segmentation results. Slightly affected by camera jitter, SAViBe+ perfectly models the dynamic background and generate better segmentation results.

From the performance in Fig.4, it follows that compared with MoG, ViBe and PKDE, SAViBe+ has better adaptability to illumination changes and disturbances from the environment.

4.2 Quantitative evaluation

A quantitative evaluation is also undertaken on the eight datasets shown in Tab.2 with the F-score and running framerate in average for all compared algorithms. The F-score measures the background subtraction accuracy by assuming both the recall and the precision, which is defined as:

$$precision = \frac{TP}{TP+FP}, recall = \frac{TP}{TP+FN}, \quad F = \frac{2 \cdot recall \cdot precision}{recall + precision}. \tag{12}$$

where *TP*, *TN*, *FP*, and *FN* are true positives (true foreground pixels), true negatives, false positives, and false negatives (false background pixels) respectively, computed in pixels of 100 successive labeled frames for each dataset. The total F-score is calculated with the *TP*, *TN*, *FP*, and *FN* summing over all the sequences for an average measure.

Table 2. Performance of F-score(%) on the complex video sequences and the running framerate(fps, frames per second) in average

scenes\methods	MoG	ViBe	PKDE	SAViBe+
backdoor	77.39	85.34	58.81	**87.66**
busStation	63.22	76.17	61.19	**77.48**
pedestrians	68.56	81.46	53.69	**84.16**
winterDriveway	57.62	62.48	56.24	**83.97**
PETS2006	63.82	80.27	63.67	**82.15**
cubicle	64.36	72.1	65.47	**75.49**
highway	56.8	79.16	55.44	**82.07**
traffic	55.71	78.82	49.83	**79.75**
F-score(total)	65.32	76.36	60.16	**82.08**
fps	31.8	**65.4**	42.7	36.9

From the F-score of each scene, it can be deduced that our method performances better than the other three state-of-the-art algorithms on all sequences. The total F-score of MoG, ViBe, PKDE and SAViBe+ are 65.32, 76.36, 60.16 and 82.08. It is evident that SAViBe+ has the highest F-score, obtaining more accurate results on average. SAViBe+ outperforms ViBe by 5.72 percent points in accuracy.

We also measured their real-time performance. The running framerate was tested on the “pedestrians” dataset with the frame size of 360 × 240. For the parameter values used in the tests, the frame rate of 31.8 fps, 65.4 fps, 42.7 fps, and 36.9 fps were achieved for MoG, ViBe, PKDE and SAViBe+ respectively, which are listed in the bottom row of Tab.2. The proposed method can run in real time.

From the subjective effect in Fig.4 and the objective data in Tab.2, it can be inferred that the performance of the proposed method is superior to the other three state-of-the-art algorithms.

5 Conclusions

In this paper, to discriminate false foreground pixels and moving object pixels, TM and HM are designed. The SAViBe+ algorithm has been proposed to enhance the ViBe algorithm with TM and HM. The proposed method detects foreground using ViBe and TM firstly, and then discriminates shadows and the moving object cascading HM with a *MofV* factor achieving high speed and good performance. Exhaustive experiments are run on eight typical datasets including complex and challenging background: indoor and outdoor scenes with shadows, scenes without shadow and scenes with camera jitter to evaluate the performance of algorithms. The experimental results prove that the proposed method enhances the performance of ViBe, compares well with other state-of-the-art methods in removing shadows, rapidly adapts to variations in the environment and operates in real-time.

There are still defects in SAViBe+. Some self shadow pixels are misclassified as background pixels when applying HM because of being darker. However, this problem can be solved by a spatial adjustment step.

Acknowledgements

This work was supported by the following funding resource: Fundamental Research Funds for the Central Universities,# K50510010007.

References:

[1] Chen Z, Ellis T, A self-adaptive Gaussian mixture model, *Computer Vision and Image Understanding*, Vol.122, 2014, pp.35-46.

[2] Joshi A J, Papanikolopoulos N P, Learning to detect moving shadows in dynamic environments, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol.30, No.11, 2008, pp.2055-2063.

[3] Tang C, Ahmad M O, Wang C, An efficient method of cast shadow removal using multiple

features, *Signal, Image and Video Processing*, Vol.7, No.4, 2013, pp.695-703.

[4] Yang Q, Tan K H, Ahuja N, Shadow removal using bilateral filtering, *Image Processing, IEEE Transactions on*, Vol.21, No.10, 2012, pp.4361-4368.

[5] Guo R, Dai Q, Hoiem D, Paired regions for shadow detection and removal, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol.35, No.12, 2013, pp.2956-2967.

[6] Zhou H, Chen Y, Feng R, A novel background subtraction method based on color invariants, *Computer Vision and Image Understanding*, Vol.117,No.11, 2013, pp.1589-1597.

[7] Brutzer S, Hoferlin B, Heidemann G, Evaluation of background subtraction techniques for video surveillance, *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* 2011, pp.1937-1944.

[8] Al-Najdawi, N., Bez, H. E., Singhai, J., & Edirisinghe, E, A. A survey of cast shadow detection algorithms. *Pattern Recognition Letters*, Vol.33, No.6, 2012, pp.752-764.

[9] Sobral, A., & Vacavant, A, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, *Computer Vision and Image Understanding*, Vol.122, 2014, pp.4-21.

[10] Elgammal A, Background Subtraction: Theory and Practice,*Wide Area Surveillance. Springer Berlin Heidelberg*, 2014, pp.1-21.

[11] Barnich, O., & Van Droogenbroeck, M, ViBe: A universal background subtraction algorithm for video sequences, *Image Processing, IEEE Transactions on*, Vol.20, No.6, 2011, pp.1709-1724.

[12] Yongqiang Li, Wanzhong Chen, Rui Jiang, The integration adjacent frame differences of improved ViBe for foreground object detection, Proceedings of 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, 2011, pp.1-4.

[13] M. Van Droogenbroeck, O. Paquot, Background Subtraction: Experiments and Improvements for ViBe, *Proceedings of 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp.32-37.

[14] Qin, Y. S., Sun, S. F., Ma, X. B., Hu, S., & Lei, B. J, A shadow removal algorithm for ViBe in HSV color space, *In 3rd International Conference on Multimedia Technology (ICMT-13)*. Atlantis Press, Nov,2013, pp.966-973.

[15] Choi, J., Yoo, Y. J., & Choi, J. Y, Adaptive shadow detector for removing shadow of moving object, *Computer Vision and Image Understanding*, Vol.114, No.9, 2010, pp.1017-1029.

[16] Spampinato C, Palazzo S, Kavasidis I, A texton-based kernel density estimation approach for background modeling under extreme conditions, *Computer Vision and Image Understanding*, Vol.122, 2014, pp.74-83.

[17] Yoshinaga S, Shimada A, Nagahara H, et al, Object detection based on spatiotemporal background models, *Computer Vision and Image Understanding*, Vol.122,2014, pp.84-91.

[18] Maddalena, L., & Petrosino, A, The< i> 3dSOBS+</i> algorithm for moving object detection, *Computer Vision and Image Understanding*, Vol.122, 2014, pp.65-73.

[19] Varcheie P D Z, Sills-Lavoie M, Bilodeau G A, A multiscale region-based motion detection and background subtraction algorithm, *Sensors*, Vol.10, No.2, 2010, pp.1041-1061.

[20] Liao S, Zhao G, Kellokumpu V, et al, Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes, *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on. 2010, pp.1301-1306.

[21] Choi J M, Chang H J, Yoo Y J, et al. Robust moving object detection against fast illumination change, *Computer Vision and Image Understanding*, Vol.116, No.2, 2012, pp.179-193.

[22] Stander, J., Mech, R., & Ostermann, J, Detection of moving cast shadows for object segmentation, *Multimedia, IEEE Transactions on*, Vol.1, No.1, 1999, pp.65-76.

[23] Heikkilä M, Pietikäinen M, Heikkilä J, A texture-based method for detecting moving objects, *BMVC*. 2004, pp.1-10.

[24] S.J. Koppal, S.G. Narasimhan, Appearance derivatives for isonormal clustering of scienes, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol.31, 2009, pp.1375-1385.

[25] A video database for testing change detection algorithms:
http://wordpress-jodoin.dmi.usherb.ca/dataset2014/