Real Time Foreground-Background Segmentation Using a Modified Codebook Model

Conference Paper · October 2009

DOI: 10.1109/AVSS.2009.85 · Source: IEEE Xplore

3 authors:

Atif Ilyas
Université Lumiere Lyon 2

Mihaela Scuturici
Université Lumiere Lyon 2

Serge Miguet
Université Lumiere Lyon 2

Real Time Foreground-Background Segmentation Using a Modified Codebook Model

Atif Ilyas, Mihaela Scuturici, Serge Miguet
Université de Lyon, CNRS
Université Lumière Lyon 2, LIRIS, UMR5205
5, Av. Pierre Mendès-France, 69676, Bron, France
{Atif.Ilyas, Mihaela.Scuturici, Serge.Miguet}@univ-lyon2.fr

***Abstract*—Real time segmentation of scene into objects and background is really important and represents an initial step of object tracking. Starting from the codebook method [4] we propose some modifications which show significant improvements in most of the normal and also difficult conditions. We include parameter of frequency for accessing, deleting, matching and adding codewords in codebook or to move cache codewords into codebook. We also propose an evaluation method in order to objectively compare several segmentation techniques, based on receiver operating characteristic (ROC) analysis and on precision and recall method. We propose to summarize the quality factor of a method by a single value based on a weighted Euclidean distance or on a harmonic mean between two related characteristics.**

***Keywords*-Segmentation; codebook; mixture of gaussions; modified codebook;**

I. Introduction

Extracting moving objects from sequences of images/ video is one of the most interesting, well focused and well addressed but still challenging topic in computer vision. Results of segmentation depend upon the variation of local or global light intensities, shadow of objects and background changes. A lot of work has already been done on background estimation, background modeling and background subtraction.

M. Pic et al. [1] use an adaptive technique for the estimation of the background on the base of learning. But learning rate is calculated after every frame. Their algorithm is computationally expensive. It fails to provide good results in the presence of fast changes in foreground and background. G. Gordon et al. [2] also use the background estimation technique on the base of distance and color. B. Han et al. [3] model the background by sequential density approximation and each time step, densities are estimated and Gaussian component is assigned to each model. K. Kim et al. [4] perform image segmentation by using codebook method. This technique shows good result and is also more robust to problems of shadow and light intensity variation. One of the more frequent use technique is modeling of background using Mixture of Gaussian [5]. It gives good results but it suffers from the problem of shadow and it is more sensitive to variation of light intensities. To avoid the complexity of computation they used the same variance for (R, G, B) color channels. A. Elgammal et al. [6] also discuss in more details and claim good results about background modeling using normalized Mixture of Gaussians. N. Thome et al. [7] also use Mixture of Gaussian and combine it with the Salvador [8] technique for shadow removal. Their results are better than the results of [5]. P. Dickinson et al. [9] model the background by an adaptive mixture of Gaussians in color and space and they claim better results than traditional mixture of Gaussians. J. Zhong et al [10] segment the foreground from the textured dynamic background by using the Kalman filter. N. Verbeke et al. [11] use a principal component analysis technique, accumulate the the last ten frames and use a technique to find the area where the motion has taken place. But the method fails if the object stops its motion and it is sensitive to changes in light intensity, shadow and sensors noise. G. L. Foresti et al. [12] use derivative model technique for detection of motion in multi cameras environment. They use both static and moving cameras in a surveillance system. Adaptive threshold technique is used to isolate objects from background. P. L. Rosin [13] compare different techniques of thresholding for change detection. They also propose segmentation-evaluation method for global image thresholding for change detection.

II. Segmentation Techniques

We decided to focus on two approaches: normalized Mixture of Gaussians (MOG) described in [7] and codebook (CB) method [4]. Indeed, these two classes of methods are considered as good ones for the background representation for real time object tracking. We apply MOG and CB on different videos having different illumination conditions like sunny day, cloudy environment and more or less moving objects. We observe that most of the time detection performances of CB are better than MOG. There are some situations where both methods fail to give satisfying performances:

- When no initial training on the empty scene is available
- When a large number of objects are moving in the scene
- When color of objects is similar to the background

A. Mixture of Gaussians

Mixture of Gaussians (MOG) technique consists of modeling a background by using N Gaussian distributions based on recent history of each pixel. The probability density function (PDF) of color value of each pixel can be formulated by using the the general equation

$$P_r(X_t = \frac{1}{N}\sum_{i=1}^{N} w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

where N is the number of Gaussian distributions, $w_{i,t}$ is an estimated weight of each PDF and η is a Gaussian probability density function.

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2pi)^{\frac{n}{2}}\left|\Sigma_{i,t}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma_{i,t}^{-1}(X_t-\mu_t)} \quad (2)$$

In (2), $\mu_{i,t}$ is mean and $\Sigma_{i,t}$ is the covariance matrix. We assume here that color channels R, G and B are statistically independent and have different color variance [7]. The weight $w_{i,t}$ of each Gaussian distribution can be calculated by using formula.

$$w_{i,t} = (1-\epsilon)w_{i,t-1} + \epsilon(M_{i,t}) \quad (3)$$

Where as ϵ is learning rate. $M_{i,t}$ is 1 for the model to whom it is matched and 0 for others. The more useful detail is available in [5]and [6], covering all the information regarding parameter updating, tunning and matching. Clever tuning of MOG is really important and also technical. If the model updates too quickly, the system takes more time to absorb changes in light intensities.

B. Codebook

Codebook (CB) method builds a background model. It is based on long scene observation and for each pixel one or several codewords will be inserted in the codebook. The number of codewords for a pixel depend upon the background variation. That is why all pixels don't have same number of codewords.

Each codeword is represented by a RGB vector v_i= $(\bar{R}, \bar{G}, \bar{B})$ and a hex-tuple $aux_i = \left\langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \right\rangle$ where $\check{I}_i$ = max$\left\{I, \check{I}_i\right\}$ and $\hat{I}_i$ = min$\left\{I, \hat{I}_i\right\}$ are the minimum and the maximum brightness assigned to each codeword respectively. f is the frequency or number of times that codeword repeats λ is the maximum negative run-length, meaning the largest time span in which this codeword is not updated/accessed. p and q are the first and the last access times of the codeword respectively.

In initial time period codewords in the codebook are created or updated using the following criteria. They use the formula of color distortion δ as

$$colordist = \delta = \sqrt{\left|\left|x_t\right|\right|^2 - C_p^2} \quad (4)$$

where C_p^2 is the autocorrelation of R, G and B colors of input pixel and the codeword, normalized by brightness.

$$C_p^2 = \left|\left|x_t\right|\right|^2 cos^2\theta = \frac{(R_iR + G_iG + B_iB)^2}{R_i^2 + G_i^2 + B_i^2} \quad (5)$$

Similarly according to [4], value of brightness $I = \sqrt{R^2+G^2+B^2}$, has two bounds: $I_{low} = \alpha\hat{I}$ is lower and I_{hi} = min$\left\{\beta\hat{I}, \frac{\check{I}}{\alpha}\right\}$ is upper limit, which are defined during codeword updating.

If $colodist \leq \Delta$ and brightness $(I_{low} \leq I \leq I_{hi})$ = true then matched codeword is updated by the method explained in [4]. According to our experiments, the value of α is between 0.7 to 0.8 and β is 1.15-1.25. If values are chosen closer to to 1, detection capabilities are increased but with an important sensibility to noise and also detect shadows as foreground like Mixture of Gaussians and other methods.

The scene can change after the initial training time, for instance, on a street surveillance application cars might enter or leave a parking, etc. If codebook is not adaptive then it will detect false background or foreground pixels due to changes in scene. To avoid this problem K. Kim et al. [4] introduce cache book. Cache words in cache book have the same structure as codewords. After the training period, if an incoming pixel matches any codeword in the codebook, then this codeword is updated. Else this pixel information is put in cache word and this pixel is treated as a foreground pixel. If any cache word is matched more frequently then put this cache word into codebook. Although the original codebook is a robust background modeling technique, there are some situations were it fails. For example, in winter, peoples commonly use black coats. If foreground-background segmentation is done using the codebook method, it adopts black color as background for many pixels. That is why many pixels are incorrectly segmented. Second, if an object in the scene stops its motion, then it is absorbed in the background. The authors indicate parameter tuning to overcome this problem, but these modifications reduce the global performance of the algorithm in other situation.

C. Modified Codebook

In this section, we suggest some changes to the original codebook algorithm. The maximum negative run length λ alone is not sufficient for filtering of codewords in the codebook. Similarly the criterion to move a cache codeword into the codebook if it stays enough time in cache is also insufficient, because these parameters are used to delete or add codewords in codebook. In the last paragraph of previous section, we discuss this problem when codebook is unable to give good results. On the basis of observation and experiments, we come to conclusion, that we should also include the parameter f into the algorithm for the accessing, deleting, matching and adding a codeword in the codebook. Similarly to move cache codewords into codebook, frequency parameter f_c of accessing this cache

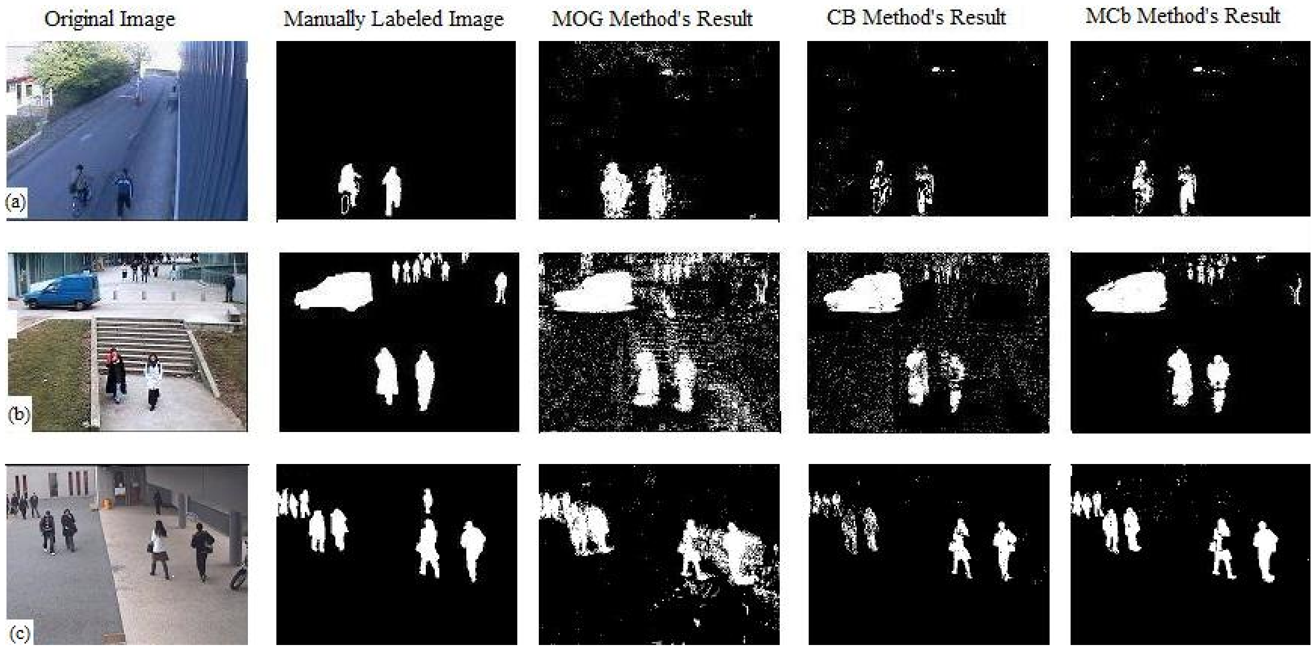

Figure 1. Original image, manually labeled image and result of three techniques are shown respectively

codebook along with reference cache codeword stay time can produce better results. Finally we add the parameter of frequency for codebook and for cache codewords as follows:

- Only codewords whose parameters of maximum negative run length $\lambda \leq \lambda_{ref}$ and code access frequency $f \geq f_{ref}$ are included in the process of matching.
- In the matching process if a new pixel value doesn't fulfill the criteria of color and brightness then this pixel is put in cache as described in [4] and marked as object pixel.
- If some cache word is staying in cache more than some reference time t_{ref} and frequency of cache word $f_c \geq f_{refc}$ then this cache word is put into codebook.

In Fig. 1(a), MOG detects objects more precisely. Where as CB is unable to detects objects well and MOG detect many background pixels as object. Which raise several problems to next step of objects recognition or tracking. in Fig. 1(b), sudden change of light intensity occurs. MOG was unable to absorb this quick change, therefore it detects more background pixels in this case as foreground pixels. But CB and MCB are more robust with this problem and MCB shows better detection rate than CB. Fig. 1(c) shows that MCB correctly detects objects with better precision than CB and MOG. It is also clear from Fig. 1, that MOG detects shadows, while this problem is not present in the CB and MCB. But once again MCB shows better objects detection. Similarly from Fig. 1 it is evident that without including the parameter of frequency f, in many situations, most of the object pixels are added to background. For example some colors are very common in object pixel like black and white. If objects having almost same or near the same color value are frequently moving in the scene, the corresponding pixel values are added into codebook and marked as background. Higher value of f also marks background pixels as foreground. Similarly, very small ranges of t_{ref} and f_c add almost all of the stationary foreground objects or moving very slowly into background.

Optimal selection of the value of these parameters is important. The value f_{refc} has to be greater or equal to f_{refc} ($f_{refc} \geq f_{ref}$). According to our experiments, value of f_{ref} and f_{refc} is between 15 and 20, value of λ_{ref} is between 150 and 200 and t_{ref} is between 40 and 60. We find these values by maximizing the result of MCB and evaluate it through techniques explained in next section. Frequency f of codeword in codebook and frequency f_c of cache word in cache book is the basic difference between the results of CB and MCB.

III. Evaluation of Segmentation Algorithms

In general, results of segmentation techniques are presented by showing some segmented video frames by using proposed algorithm and some standard techniques. But this method is not sufficient for evaluation. Some of authors use evaluation techniques which are based on receiver operating characteristic (ROC) to show the comparison between different techniques. Some useful work is available in [14], [15]. They use specificity and sensitivity to plot the graph of segmentation techniques. We use precision and recall as well to plot the graph and give also results in the form of a single number. Our method consist of five steps:

1) Select different frames from several challenging videos including high variation in light intensities, large numbers of moving or stationary objects, sun light, clouds etc. Some of these image frames are shown in Fig. 1.
2) manually Label selected frames. These are used as ideally segmented reference frames.
3) Calculate the true positive TP, false positive FP, true negative TN and false negative FN by comparing ideal segmented frames with segmented frames using CB, MCB and MOG. Where as TP: means ground truth and system result both agree that pixel belongs to object. FP: means ground truth declare this pixel belongs to background but system result is false. TN: ground truth and system result both agree that pixel belongs to background FN: system results tell that pixel is a part of background but ground truth is against of it.
4) Calculate false positive rate (FPR), true positive rate (TPR), precision (PR) and recall (RE) of each stored frame for each pixel.

$$TPR = RE = \frac{TP}{TP + FN}$$

$$FPR = 1 - \frac{TN}{TN + FP}$$

$$PR = \frac{TP}{TP + FP}$$

5a) Precision and recall may be combined into a single statistic number F. Which is harmonic mean of precision and recall. Harmonic mean is an appropriate representation for situations when average of rates is desired.

$$F = 2\left(\frac{PR * RE}{PR + RE}\right) \qquad (6)$$

5b) We also propose a formula, which is a weighted Euclidean distance that can be adapted to the needs of the application.

$$E = \sum_{j=1}^{m} \sqrt{\gamma\,(FPR)^2 + (1-\gamma)\,(1-TPR)^2} \qquad (7)$$

Where E is the sum of all the weighted distance from the ideal position to calculated position of TPR and FPR. γ is a weighting coefficient of error from ideal position. Selection of γ parameter is discussed more in detail in section IV.

5c) We also compare segmentation methods by using other evaluation techniques which are used by [13]. These parameters are the percentage of correct classification (PCC) and Jaccard coefficient.

$$PCC = \frac{TP + TN}{TP + FN + FP + TN}$$

$$JC = \frac{TP}{TP + FP + FN}$$

IV. Detection Results and Comparison

In this section we discuss the results of segmentation techniques explained in section II on the basis of the evaluation methodology discussed in section III. In general, large and fast changes in light intensities cause instability in all segmentation techniques. Most sensitive segmentation techniques (like MOG) produce large variance. That becomes the reason for low probabilities of particular event. MOG is more computationally expensive than CB [4] and modified CB. MCB shows its ability to process more frame/sec as compared to CB and MOG. We calculate frame rate on laptop having a Core Duo processor 1.86 GHz. We present results in Tab. 1. We don't use synthetic data, which is easy to evaluate but rarely true indicator of real scenario. We used five videos of fifteen to fifty five minutes. We select these videos which have to different illumination conditions and one to many foreground objects. We don't have an initial training set on an empty scene. In performance measurement technique, we take round 50 frames of indoor and outdoor videos. These frames are selected after 200 frames to provide enough time to model the background. We calculate TPR, FPR, RE and PR for each of frame. We also calculate precision quality factor F using (5) and add results of all the frames.

In Fig. 2, we plot Recall (RE) as a function of precision (PR). Each point of the graph represents the result for a single frame. The ideal value of the PR=1 and RE=1 (top right corner). MOG has good recall but has a worse precision as compared to CB and MCB. Only 4% of frames have precision greater than 0.8. CB has good precision, about 20%, but very low recall. MCB shows highest value i.e 32% of frames having precision more than 0.8 and a recall comparable to MOG. In Tab. 1, value of F for MOG, CB and MCB are 28.28, 24.15 and 32.17. Highest number for MCB verified our claim also.

Our second series of experiments use step 5b evaluation method. We first plot the FPR and TPR which are also known as specificity and sensitivity respectively (Fig. 3). Each point is actually representing FPR and TPR of one frame. The ideal values of FPR=0 and TPR=1 (top left corner). In a real life scenario, the quantity to optimize

Table I
Comparison of different methods

Method	*MOG*	*CB*	*MCB*
Precision quality factor (F)	28.28	24.15	**32.17**
FPR based error factor (E) for γ=0.95	3.02	5.34	**2.87**
Euclidean distance based on specificity	**9.07**	22.40	11.74
Percentage of error coefficient (PCC)	48.04	48.02	**48.95**
Jaccard Coefficient (JC)	20.75	17.56	**25.22**
Segmentation rate (frame/sec)	11.10	12.79	**13.48**

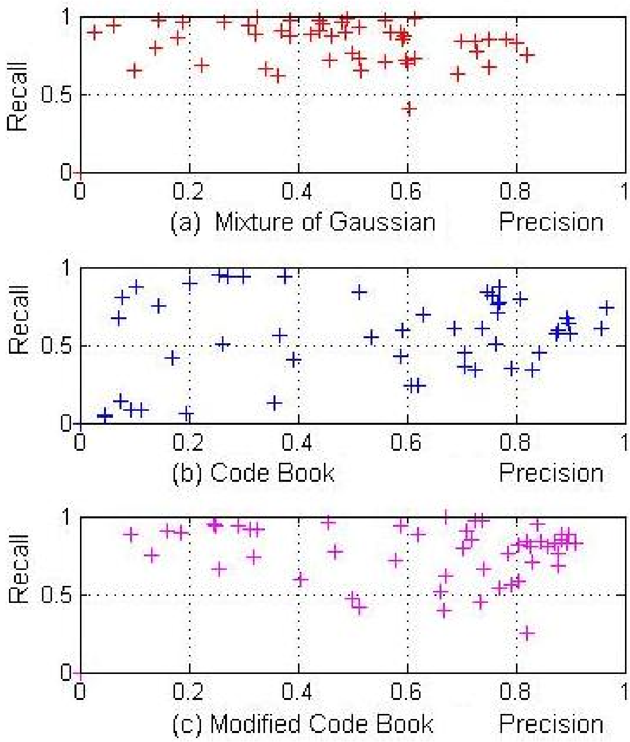

Figure 2. Precision and recall of different segmentation techniques

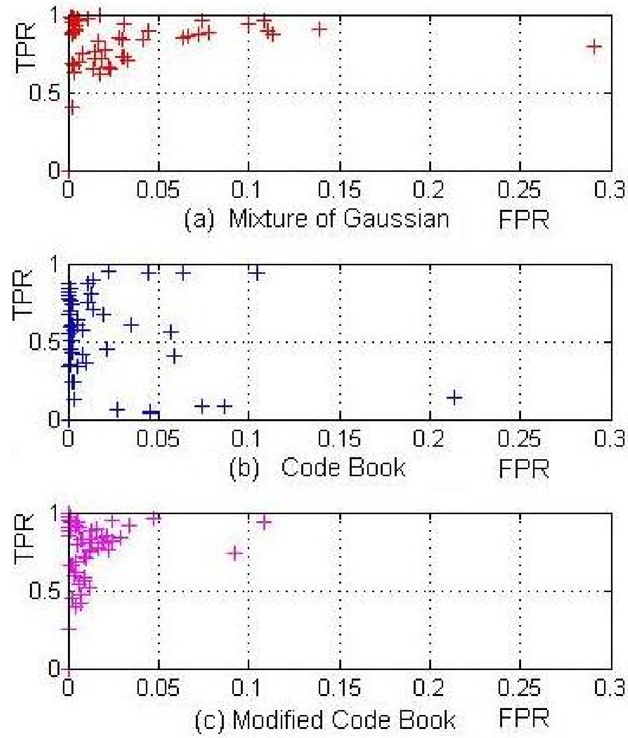

Figure 3. True positive and false positive rate of different segmentation techniques

depends upon the application. If we want to minimize the false alarms then FPR should be minimized but if we don't want to miss any foreground object then FPR can be adjusted in order to maximize TPR. Normally FPR also increase when TPR increase [14, 15, 16, 17]. If we increase the sensitivity of detection, then more noise and light variation will be induced in segmented scene.

The expected result is always a trade off between sensitivity and specificity. In some video surveillance applications we don't afford too much false alarms. For example during weekend, maybe we want to observe some regions but we can't tolerate more than 5% false alarms. In this case we are compromising to ignore some moving objects. From Fig. 3, it is clear that if we don't want more than 5% false alarm then MCB is better than CB and MOG because it have two false positive and it has good sensitivity also. In this case you can use the higher value of $\gamma \geq 0.95$ in (7). The higher value of γ makes (7) more influenced by FPR. Value of E is higher for the algorithm that has more FPR. In Tab. 1 the smallest error, for E with γ=0.95, is obtained by MCB algorithm, followed by MOG and CB. In prohibited or restricted area, we can afford large number of false alarms but we can't afford any object to be missed by the system. In this case it is better to select MOG because of their sensitivity that is better than CB.

Result of PCC and JC in Tab. 1 also verify our claim that MCB is a better segmentation technique than CB and MOG.

V. Conclusion and Discussion

In section IV we discussed results of MOG, CB and MCB. We have included parameter of frequency f and f_c in the codebook algorithm for the accessing, deleting, matching and adding codeword in the codebook or to move cache codewords into codebook. Comparing MCB with two other techniques CB and MOG shows that MCB is able to produce better results. In short we can summarize the results as follow:

1. In [4], it is claimed that CB works better than MOG and other techniques. In our observation, this is the case only when few objects with less similarity of color with background are present.

2. MCB introduces no new parameter to the original CB. It uses the parameter of frequency for improvement of CB. It doesn't introduce any additional computational complexity so it is still computationally less expensive than MOG. But it is able to detect moving object more precisely than codebook and probability based mixture of Gaussians.

3. We evaluate our results in three ways. Visual (qualitative) results can be appreciated on images. ROC analysis plots the performances either in terms of precision versus sensitivity or in the terms of TPR and FPR. At last, two methods for computing a unique quality factor are given. The three methodologies indicate that our proposed MCB shows better result than MOG and CB. One can see from the Tab. 1

that MCB over performs CB in precision quality, specificity based error factor as well as weighted Euclidean distance. In comparison with MOG, MCB obtains better precision factor, less false alarms and is more robust with variation of light intensities. Moreover, it involves less floating point calculations.

From the above discussion we can conclude that MCB can work better in almost all the conditions as compared to CB, without introducing more complex calculations in the algorithm. The choice between MCB and MOG depends on the application. If the precision is considered as the most important factor, then MCB is probably the best choice. If a compromise is possible on precision, shadow and false alarms but any small object shouldn't be lost, then MOG can be a better choice. Nevertheless, like CB and MOG, MCB doesn't handle the case of still objects in a satisfying manner. If foreground object stops for some time, then it will also be included in the background. If f_c is increased to avoid this problem, then it takes more time to be absorbed in the background, but noise level is increased. A more sophisticated algorithm based on object recognition might be useful to overcome this problem.

References

[1] M. Pic, L. Berthouze and T. Kurita, "Adaptive Background Estimation: Computing a Pixel-Wise Learning Rate from Local Confidence and Global Correlation," IEICE Transaction Information and System, vol. E-87D, pp. 50-57, 2004.

[2] G. Gordon, T. Darrel, H. Harville and J. Woodfill, "Background Estimation and Removal Based on Range and Color," Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 459-464, 1999.

[3] B. Han, D. Comaniciu and L. Davis, "Sequence Kernel Density Approximation through Mode Propagation: Application to Background Modeling," Trans. on Pattern Analysis and Machine Intelligence, vol. 30, pp. 1186-1197, 2008.

[4] K. Kim, T. Thanarat, H. Chalidabbhognse, D. Harwood and L. Davis, "Real Time Foreground-Background Segmentation using Codebook Model," Real-Time Imaging, vol. 11, pp. 172-185, 2005.

[5] C. Stauffer, W. Eric and L. Grimson, "Learning Patterns of Activity Using Real Time Tracking," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, pp. 747-757, 2000.

[6] A. Elgammal, D. Harwood and L. Davis, "Non-Parametric Model for Background Subtraction," 6th European Conference on Computer Vision, Lecture Notes In Computer Science (LNCS), vol. 1843, pp. 751-767, 2000.

[7] N. Thome and S. Miguet, "A Robust Appearance Model for Tracking Human Motion," Advanced Video and Signal Based Surveillance AVSS, pp. 528-533, 2005.

[8] E. Salvador, P. Green and T. Ebrahimi, "Shadow Identification and classification Using Invariant Color Model," Proceeding of IEEE International conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, pp. 1545-1548, 2001.

[9] P. Dickinson, A. Hunter and K. Appiah, "A Spatially Distributed Model for Foreground Segmentation", Image and Vision Computing, December 2008.

[10] J. Zhong and S. Sclaroff, "Segmentating Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter," IEEE International Conference on Computer Vision, vol. 1, pp. 44-50, 2003.

[11] N. Verbeke and N. Vincent, "A PCA-Based Technique to Detect Moving Objects," Scandinavian Conference on Image Analysis (SCIA), LNCS 4522, pp. 641-650, 2007.

[12] G. Foresti, C. Micheloni, L. Sindaro, P. Remagnino and T. Ellis, "Advanced Image and Video Processing in Active-Based Surveillance System," IEEE Signal Processing Magn., vol. 22, pp. 25-37, 2005.

[13] P. L. Rosin and E. Ioannidis, " Evaluattion of Global Image Thresholding for Change Detection," Pattern Recognition Letter, vol. 24, pp. 2345-2356, 2003.

[14] T. H. Chalidabhongse, K. Kim, D. Harwood and L. Davis, "A Perturbation Method for Evaluating Background Subtraction Algorithms," Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillances (VSPETS), October, 2003.

[15] J. Davis and M. Goadrich, "The Relationship between Precision-Recall and ROC Curves," ACM Proc. of the 23rd International Conference on Machine Learning (ICML), vol. 148, pp. 233 - 240, 2006.

[16] Ji. Wang, Ze Wang, Geoffrey K. Aguirre, and J A. Detre, "To Smooth or Not to Smooth? ROC Analysis of Perfusion fMRI Data," Magnetic resonance imaging, vol. 23, pp. 75-81, 2005.

[17] H.S. Sheshadri, and A. Kandaswamy, "Computer Aided Decision System for Early Detection of Breast Cancer," Indian Journal of Medical Research, vol. 124, pp. 149-154, Aug 2006.