

Foreground Segmentation in Video Sequences with a Dynamic Background

Chu Tang, M. Omair Ahmad, and Chunyan Wang
Department of Electrical and Computer Engineering
Concordia University, Montreal, Canada

Abstract - Segmentation of a moving foreground from video sequences, in the presence of a rapidly changing background, is a difficult problem. In this paper, a novel technique for an effective segmentation of the moving foreground from video sequences with a dynamic background is developed. The segmentation problem is treated as a problem of classifying the foreground and background pixels of a video frame using the color components of the pixels as multiple features of the images. The gray levels of the pixels and the hue and saturation level components in the HSV representation of the pixels of a frame are used to form a scalar-valued feature image. This feature image incorporating multiple features of the pixels is then used to devise a simple classification scheme in the framework of a support vector machine classifier. Unlike some other data classification approaches for foreground segmentation in which *a priori* knowledge of the shape and size of the moving foreground is essential, in the proposed method, training samples are obtained in an automatic manner. In order to assess the effectiveness of the proposed method, the new scheme is applied to a number of video sequences with a dynamic background and the results are compared with those obtained by using other existing methods. The subjective and objective results show the superiority of the proposed scheme in providing a segmented foreground binary mask that fits more closely with the corresponding ground truth mask than those obtained by the other methods do.

Keywords - Foreground segmentation; support vector machine (SVM); spatial-temporal feature; dynamic background

I. INTRODUCTION

Segmenting a moving foreground from a video sequence is a first step in object segmentation. There exist many segmentation algorithms in the literature. In [1], Stauffer and Grimson have proposed an algorithm based on the assumption that a background pixel has a Gaussian mixture model (GMM) distribution. This method can, to some extent, deal with slowly changing background. ViBe [2] is another background subtraction method. It uses a set of values taken in the past at the same location or in the neighborhood, 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.

In many scenes especially in outdoor ones, the background can be rapidly changing, e.g., scenes with waving water or swaying tree branches, making the foreground segmentation more challenging. In [3], a background subtraction method, in which the background

is modeled using texture features, extracted by using a modified local binary pattern (LBP), has been proposed for segmenting the foreground. In [4], on the premise that the LBP method does not take into account the motion information of the moving objects in a video, an approach of texture pattern flow (TPF) is used to model each pixel to build the background model. In [5-7], two-stage schemes have been devised for foreground segmentation. The first stage is for a coarse segmentation and then the results are refined in the second stage. In [5], the authors have used an alarm trigger module in the second stage to detect the background pixels using the spatial and temporal features of the scene. The method requires a pre-specified threshold parameter for its alarm trigger module, which does not universally apply to all kinds of objects. In the methods of [6] and [7], the foreground segmentation for the second stage is considered as a data classification problem and the technique of a support vector machine (SVM) on the spatiotemporal features of the image scenes is applied. However, the methods of [6] and [7] require a manual selection of the training points and the former is specifically designed for the segmentation of rectangular vehicle objects.

In this paper, a novel two-stage scheme for foreground segmentation is presented. It involves GMM and SVM. With regard to the second stage, the proposed method has two distinct novelties. First, unlike the methods of [6] and [7] that require some a priori knowledge of the scene object for the selection of training samples for SVM, the proposed scheme develops a mechanism for an automatic selection of training samples. Since the mechanism for the selection is based on some motion information of the foreground, it can adapt to objects not only of different kinds of shapes, but also to objects with their shapes changing frame to frame. Second, a spatiotemporal feature image that can more accurately distinguish different kinds of moving foreground and background is proposed to distinguish more effectively the foreground pixels from those of the background.

II. PROPOSED METHOD

In the proposed method, the segmentation of the foreground from a sequence is obtained in two stages: GMM is used in the first stage to remove most of the background pixels from the scene; in the second stage, an SVM-based segmentation scheme is applied to the foreground obtained from the first stage using the information from the outputs of the first stage as well as that from the original sequence in order to get a final foreground. A schematic of the proposed method is shown in Fig. 1.

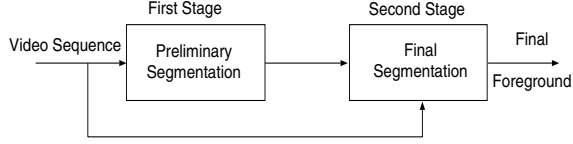


Figure 1 Two stages of the proposed segmentation scheme

A. GMM-Based Foreground Segmentation

To perform a preliminary foreground segmentation in the first stage, the GMM method is adopted. In this method, a pixel in the current frame is examined, using the background model of the previous frame, to determine whether it belongs to the foreground or the background of the current frame, while at the same time, a background model for the current frame is also built pixel by pixel.

Since the GMM method integrates in it the information on the pixel history, it can effectively deal with a slowly-changing background. On the other hand, for scenes with a rapidly-changing background, a significant amount of background pixels are detected as foreground ones. Fig. 2 is an example of the results of segmenting a foreground from a video sequence with a dynamic background by using the GMM method. Fig. 2(c) shows that there are a large number of pixels belonging to the dynamic background that are falsely detected as foreground pixels, thus making a further segmentation of the foreground a necessity in order to remove these moving background pixels depicted by Fig. 2(c).

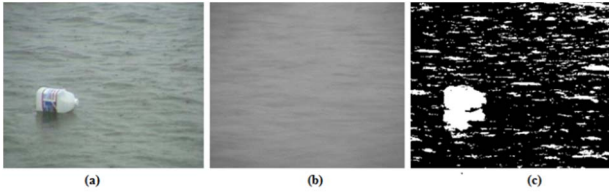


Figure 2 An example of foreground segmentation using the GMM method [1]. (a) Original frame. (b) Gray level background image obtained by using the GMM method. (c) Binary foreground mask obtained by using the GMM method.

B. Data Classification-Based Foreground Segmentation

In the second stage, the objective is to remove the moving background pixels from the segmented foreground obtained from the first stage. The main idea used in the second stage is to treat the segmentation problem of this stage as a classification of the foreground pixels and those of the moving parts of the background.

In the proposed method, a spatiotemporal feature image using multiple features of the image is constructed in order to make the difference between the foreground and background pixels more pronounced for classification. To this end, a small but equal number of image pixels called the training samples are selected from each of the two classes. The image pixels with their features specified by the feature image are then classified using classification knowledge of the training samples in a support vector machine (SVM) classifier. In the proposed scheme, a method is also developed for an automatic generation of the training samples. A more detailed schematic containing different modules to perform the various tasks

of the proposed scheme for foreground segmentation is shown in Fig. 3.

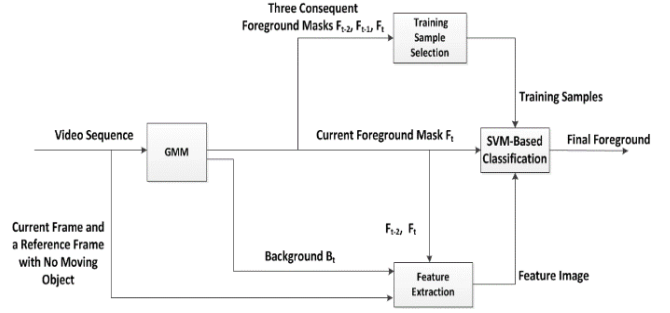


Figure 3 Detailed architecture of the proposed method.

2.2.1 Selection of Training Sample

In order to train a classifier, certain number of samples, each labeled with its class, is needed. In the proposed scheme, a selection of the training samples is carried out automatically by the Training Sample Selection module, without *priori* knowledge of the foreground/background. For the purpose of this selection, we use the segmented foreground masks obtained from the GMM module.

As seen from Fig. 2(c), the size of the cluster of pixels representing the actual foreground is larger than those corresponding to the moving background pixels. This observation is used to obtain a mask that consists of pixels that predominantly belong to the foreground. Such a mask can be obtained as

$$F_m = F_{t-2} \bullet F_{t-1} \bullet F_t \quad (1)$$

where F_{t-2} , F_{t-1} and F_t denote the foreground masks obtained from the GMM module corresponding to the current and previous two frames, and \bullet represents pixel-wise AND operation.

A morphological opening operation [8] is then be applied to the mask F_m in order to further remove the background pixels from it and to obtain the final mask F_{om} that consists overwhelmingly of the pixels representing the foreground. If K_F is the number of foreground training samples to be chosen from F_{om} , and N_F is the total number of foreground pixels in F_{om} , then K_F training samples are selected by uniformly sampling the foreground region in F_{om} at a spatial sampling rate N_F/K_F , thus ensuring the foreground training samples to be evenly distributed.

Next, a technique for selecting background training samples from F_t is also developed. In order to ensure that all the samples in this selection are only the moving background pixels, a sufficiently large region R of F_t that possibly contains all the foreground pixels needs to be excluded from F_t before this selection. To do so, the mask F_{om} is divided into blocks of an appropriate size, and the region R is obtained as a polygon consisting of a contiguous set of blocks such that: (i) none of the peripheral blocks in R has logic “1” pixel, and (ii) each of the blocks interior to the peripheral blocks have at least one logic “1” pixel. With an appropriate choice of block size, this method should ensure that the region R covers all the foreground pixels even if some of these pixels are not identified in F_{om} .

Once the region R has been identified in F_{om} , all pixels of F_t within R are made to have a logic “0” value, giving a

mask F_b . From this mask F_b , K_B pixels are selected as background training samples by uniformly sampling the pixels with logic "1" in F_b .

2.2.2 Feature Extraction

Data is usually classified based on the differences in a feature or features of the data belonging to the different classes. As mentioned earlier, for the present problem, there are only two classes: moving foreground pixels and those belonging to moving background. The objective of the feature extraction module of the proposed scheme is to construct a feature image based on multiple features such that with reference to such a feature image, the moving background pixels are significantly different from those belonging to the moving foreground.

Let I_G be a gray level frame and B_G the gray level background image corresponding to I_G . The value in $|I_G - B_G|$ corresponding to a pixel position in the moving foreground will be, in general, larger than that corresponding to a pixel position in the background. Since it is not possible to have B_G corresponding to an I_G , we should consider the use of an approximate alternative of B_G . Fig. 4(b) shows a gray level frame of the sequence that does not happen to have the object appearing in its scene. Fig. 4(c) is the absolute difference frame. It is seen from this figure, that intensity at many of the background pixel positions is not as low as one would like to have. This is because of the fact that background frame is not the one corresponding to I_G . An alternative is to use the background frame corresponding to I_G generated by the GMM module. The background frame, shown in Fig. 4(d), has a better correspondence with I_G in terms of the moving background pixels, since it is created using the latter. Fig. 4(e) shows the difference image I_{DG} when the GMM generated background image is used. A comparison between Fig. 4(c) and (e) indicates that the use of the GMM-generated background image provides a better distinction between the moving foreground and background pixels.

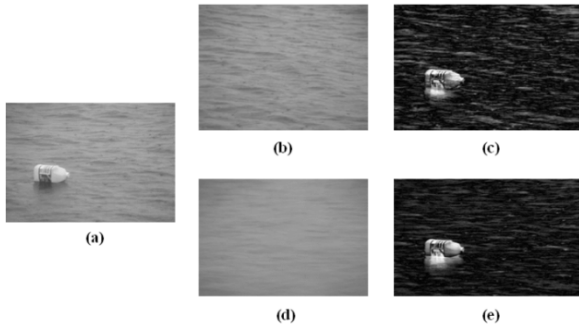


Figure 4 Generation of an alternative gray level background image. (a) Original gray level frame. (b) A gray level frame containing no object. (c) Difference between (a) and (b). (d) Gray level background image produced by GMM. (e) Difference between (a) and (d).

Next, we consider the color components of frame to construct the respective difference images. In view of the fact that it would be computationally very expensive to construct a color background image corresponding to a frame under consideration, we obtain a color background image by computing the average of the first 10 frames of the sequence.

In the proposed scheme, the color feature is considered based on the HSV system. Since the intensity has already been used in considering the gray level feature, we make use of the hue and saturation components of the color for constructing two other difference images. Fig. 5 (a) and (b) show the hue components corresponding to the current color frame and the color background image, respectively. The absolute difference hue image I_{DH} is shown in Fig. 5(c). Similarly, the saturation components of the foreground, background and absolute difference (I_{DS}) images are given by the images shown in Fig. 6(a), (b) and (c), respectively. For the sake of uniformity, the pixels in the images of Fig. 5 and 6 are re-quantized in order to have the same number of levels as the number of gray levels of the images in Fig. 5, i.e., 256.

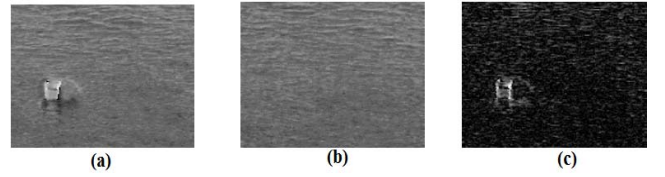


Figure 5 Hue components. (a) Hue component of the current frame. (b) Hue component of the background. (c) Difference between (a) and (b).

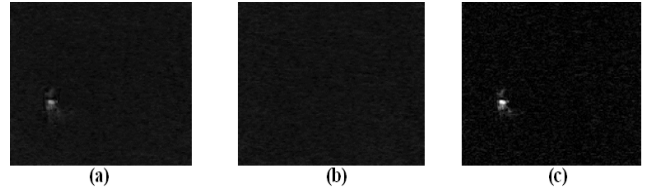


Figure 6 Saturation components. (a) Saturation component of the current frame. (b) Saturation component of the background. (c) Difference between (a) and (b).

As in the case of the gray level difference image I_{DG} , in the hue and saturation difference images I_{DH} and I_{DS} , the pixel values corresponding to the background region are, in general, smaller than that corresponding to the foreground region. This difference in the pixel values of the two regions can be further pronounced by obtaining a weighted sum of the three difference images, given by

$$I_{D0} = w_1 \bullet I_{DG} + w_2 \bullet I_{DH} + w_3 \bullet I_{DS} \quad (2)$$

where w_1 , w_2 and w_3 are the weights of the individual difference images used to obtain the overall difference image I_{D0} . The idea behind this weighted sum instead of having simply a sum of the three difference images, is to emphasize or de-emphasize the importance of a difference image in obtaining I_{D0} depending on its ability to distinct the two types of pixels. In order to ascertain the values of the three weights, we proceed as follows. Using the positions of the foreground and background samples as determined in the previous section, we determine the median pixel values for the foreground and background pixels corresponding to the position of the training samples in the three difference images and obtain the following three ratios:

$$R_G = \frac{\hat{I}_{GF}}{\hat{I}_{GB}}, R_H = \frac{\hat{I}_{HF}}{\hat{I}_{HB}}, R_S = \frac{\hat{I}_{SF}}{\hat{I}_{SB}} \quad (3)$$

where \hat{I}_{GB} , \hat{I}_{HB} and \hat{I}_{SB} are the median gray, hue and saturation levels of the background samples, and \hat{I}_{GF} , \hat{I}_{HF} and \hat{I}_{SF} are the respective median values of the

foreground samples. We have performed an experiment involving different kinds of frames and have observed that the values of R_G , R_F and R_S are approximately in the ratio 1:0.6:0.4. Accordingly, we choose the values of weights as $w_1 = 0.5$, $w_2 = 0.3$ and $w_3 = 0.2$.

Often the values of I_{D0} at the boundary of the foreground are lower than that in its interior. Thus, the expression for I_{D0} as given by (2) needs to be modified in order to avoid such a possibility. In order to rectify this problem, the temporal difference between frames is used to enhance the values of the boundary pixels of the foreground. Let IF_t and IF_{t-2} denote the two gray level foreground images produced by the GMM module corresponding to the current and previous to the previous frames. The difference image of these two frames is

$$I_{DT} = |IF_t - IF_{t-2}| \quad (4)$$

Since in I_{DT} , the values of the pixels at the boundary of the foreground region are, in general, larger than those of the non-boundary region, I_{D0} given by (2) is modified by adding to it the I_{DT} with a small weight:

$$\tilde{I}_D = w_1 \cdot I_{DG} + w_2 \cdot I_{DH} + w_3 \cdot I_{DS} + w_4 \cdot I_{DT} \quad (5)$$

The value of the weight w_4 is chosen to be smaller relative to the other weights so as not to increase the pixel values at the boundaries of the various regions representing the moving background pixels in I_{D0} . In our experiments, $w_4 = 0.1$. The values of the other weights are, therefore, modified as $w_1 = 0.45$, $w_2 = 0.27$, $w_3 = 0.18$ in order to normalize the four weights. Fig. 7 shows the overall difference image. It is seen from this figure that the contrast between the foreground and background pixels is, in general, more than in any of its constituent difference images. Since the pixel values of I_D will be used as the feature of a given frame to distinguish between its foreground and background pixels, I_D is a feature image.



Figure 7 Constructed feature image I_D .

Similar to other techniques, the proposed method makes use of multiple features. However, its main advantage lies in incorporating these multiple features into a single feature characterized by the pixel values of the feature image I_D . Regardless of the nature of the foreground or moving background pixels and the number of features used, the proposed feature extraction method results in a feature image with scalar-valued pixels. This very characteristic of I_D , as we will see in the next subsection, can be used to simplify the classification of foreground and moving background pixels.

2.2.3 SVM-Based Classification

In this section, the pixels corresponding to the foreground mask produced by the GMM module are classified using the classification technique of SVM [9]. The SVM technique for classification is known to provide good results in situations such as ours, where the number of training samples is limited [10]. In order to reduce the risk

of misclassifications, in the SVM classifier, we use the histogram of the pixels in a window centered at the pixel in question, instead of using the pixel value alone. Also, in order to better discriminate the foreground and background pixels in the histogram feature space, we apply a kernel function as described in [11].

By using the feature image I_D , the SVM module first constructs local histograms for all the pixels in the foreground mask produced by the GMM module. The pixel p is then classified as a foreground or moving background pixel using the trained SVM classifier with its extracted histogram feature.

III. PERFORMANCE EVALUATION

In order to assess the proposed method, in this section, we apply it to segment the foregrounds of a number of video sequences with a dynamic background. The visual and quantitative results are compared with those obtained by using and six other methods presented in [1] [2-6].

For our experiments, we set the number of training samples from the moving foreground and that from the moving background as $K_F = K_B = 400$. In order to reduce the computational cost for the computation of histograms, we first obtain the integral histogram of I_D , which has a linear complexity to the data length, and then compute the actual histograms quite simply, as in [12].

We conduct the experiment on three video sequences, *Water* [13], *WaterSurface* [14] and *Curtain* [14]. Based on the resolutions of the three sequences, the coefficients for the opening operation, \mathcal{E}_{erode} and \mathcal{E}_{dilate} , are set as (8, 6), (5, 3) and (5, 3), respectively. The window size for the histogram calculation in the classification module are set to 11x11, 9x9 and 9x9, respectively, for the three sequences.

The results of applying the proposed segmentation method and the methods of [2-6] on the *Water*, *WaterSurface* and *Curtain* sequences are shown, respectively, in Fig. 8, 9 and 10.

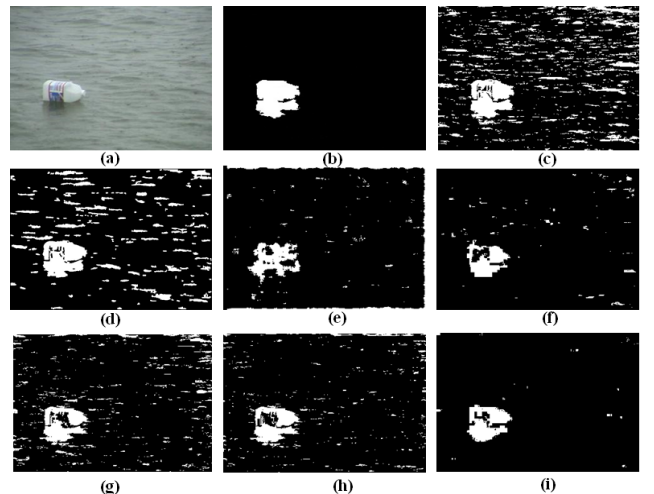


Figure 8 Segmented foreground masks obtained by applying various segmentation methods on the *Water* sequence.

(a) Original 190th frame. (b) Ground truth of the foreground. (c) GMM [1]. (d) Method [2]. (e) Method [3]. (f) Method [4]. (g) Method [5]. (h) Method [6]. (i) Proposed method.

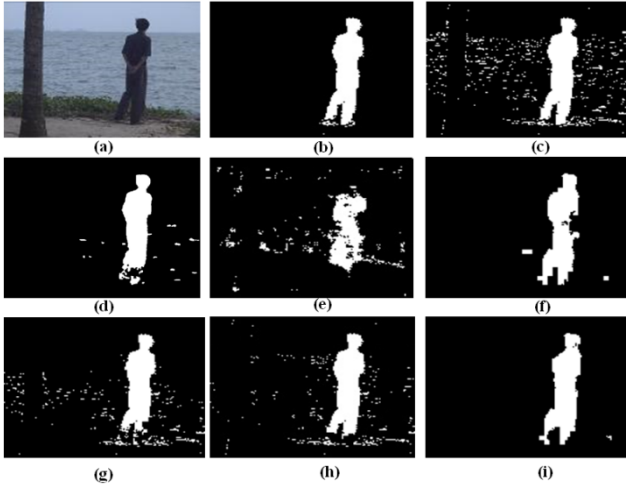


Figure 9 Segmented foreground masks obtained by applying various segmentation methods on the *Watersurface* sequence. (a) Original 578th frame. (b) Ground truth of the foreground. (c) GMM [1]. (d) Method [2]. (e) Method [3]. (f) Method [4]. (g) Method [5]. (h) Method [6]. (i) Proposed method.

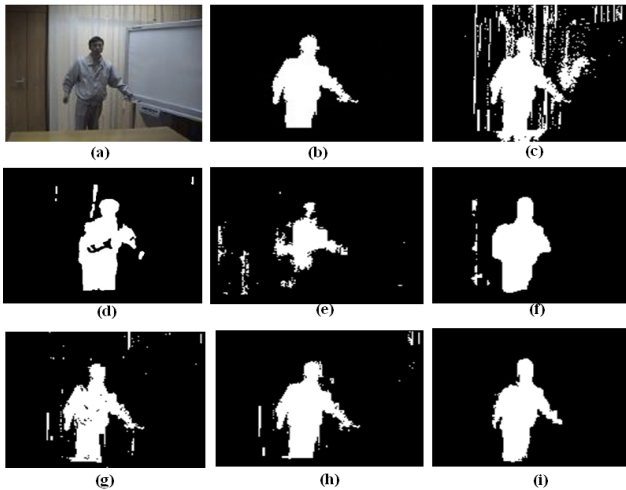


Figure 10 Segmented foreground masks obtained by applying various segmentation methods on the *Curtain* sequence. (a) Original 882nd frame. (b) Ground truth of the foreground. (c) GMM [1]. (d) Method [2]. (e) Method [3]. (f) Method [4]. (g) Method [5]. (h) Method [6]. (i) Proposed method.

Fig. 8(a), 9(a) and 10(a) show the original 190th, 578th and 882nd frames of the three sequences, whereas Fig. 8(b), 9(b) and 10(b) show the respective ground truths of the foregrounds. The images (c-h) in the three figures show the results of the foreground segmentation obtained by applying, respectively, the GMM method, the methods of [2-6], and the proposed method. It is seen from the illustrations shown in (c) of these figures that the GMM method even though segments almost all the foreground pixels, it includes in its segmentation many of the dynamic background pixels, since the method is quite sensitive to pixel motions. It is seen from illustration (d) and (e) of the figures that even though the methods presented in [2] and [3], which use neighboring pixel values and texture feature, respectively, are less sensitive to pixel motions in comparison to the GMM method, there are still a number of background pixels that are misclassified. The method in [4] gives relatively better results, as seen from its segmentation results shown in the illustrations (f) of the

figures; however, it still removes some of the foreground pixels or includes some of the pixels belonging to the dynamic background. It is seen from the illustrations (g) and (h) of Figs. 11-13 that the methods in [5] and [6], which are also two-stage schemes, produce a moving object that is more complete, compared to that produced by the methods of [3] and [4] (illustration (e) and (f)), but at the same time, a number of background pixels are also included. The proposed method, however, is seen to provide the best results in terms of the completeness of the segmented foreground and exclusion of most of the background pixels.

For quantitative evaluation of the various segmentation methods, false positive (FP), false negative (FN), false alarm rate (FAR), and tracker detection rate (TRDR) [15] are used as performance metrics.

For the objective evaluations of the methods, 25 frames are randomly selected from the set of the frames containing the moving object in each sequence. Each segmented foreground mask obtained by using a given method is compared with the corresponding ground truth in order to obtain the values of the performance metrics and averaged over the 25 frames of each sequence. Tables 1 and 2 give, respectively, the average numbers of false positives and false negatives per frame. It is seen from these tables that the proposed method provides the lowest values for the false positives, and the second lowest values, in the most cases, for the false negatives. The reason for the GMM method for providing the lowest FN values can be attributed to the fact that this method is very sensitive to pixel motions, thus classifying all the moving pixels as foreground, as indicated by the very large values of false positives produced by it.

Table 1. Average number of false positives per frame

	[1]	[2]	[3]	[4]	[5]	[6]	proposed
Water	8024	6318	3934	424	2175	1857	191
Water surf.	779	344	749	246	317	349	172
Curtain	1980	316	529	369	413	449	289

Table 2. Average number of false negatives per frame

	[1]	[2]	[3]	[4]	[5]	[6]	proposed
Water	214	233	1127	833	659	847	519
Water surf.	95	211	596	192	189	196	180
Curtain	129	281	434	274	284	268	261

Table 3. Average false alarm rate

	[1]	[2]	[3]	[4]	[5]	[6]	Proposed
Water	0.69	0.61	0.58	0.17	0.44	0.35	0.08
Water surf.	0.32	0.14	0.42	0.16	0.26	0.21	0.11
Curtain	0.41	0.14	0.24	0.15	0.23	0.19	0.09

Table 4. Average tracker detection rate

	[1]	[2]	[3]	[4]	[5]	[6]	Proposed
Water	0.96	0.94	0.73	0.75	0.77	0.74	0.86
Water surf.	0.97	0.88	0.68	0.86	0.83	0.85	0.90
Curtain	0.98	0.84	0.82	0.84	0.82	0.86	0.89

Tables 3 and 4 give, respectively, the average false alarm rate and tracker detection rate. From these tables, it is seen that the proposed method gives the lowest FAR, which means that the ratio of the falsely classified foreground pixel to the total number of pixels classified as foreground pixels is the lowest. In most of the cases, the proposed method also has the second highest TRDR. The

highest TRDR provided by the GMM method is resulted from its sensitivity to motions of the pixels thus classifying a moving pixel as a foreground pixel irrespective of whether it belongs to the foreground or background.

The computation times of the proposed method and those of [2-5] are obtained by applying these methods to the 160x128 resolution *WaterSurface* and *Curtain* sequences on a Windows-platform PC with a 2.83 GHz Intel Core Quad CPU and 8 GB RAM using MATLAB codes. The results are shown in Table 5, in which the computation time of method [6] is not included, since it needs a manual sample selection and an off-line training procedure for SVM. The results presented in this table show that the proposed method on an average takes 87%, 44% and 37% more time than the methods of [2], [3] and [5], respectively, with a significant superior segmentation. In comparison to the method given in [4], the proposed method not only provides a superior performance but also a computation time that is 10% lower.

Table 5. Average computation time per frame (Second)

		[2]	[3]	[4]	[5]	Proposed
Sequence	<i>WaterSurface</i>	5.70	7.52	11.75	7.78	10.38
	<i>Curtain</i>	4.63	5.81	9.59	6.24	8.79
Average		5.12	6.67	10.67	7.01	9.59

IV. CONCLUSION

An accurate segmentation of moving foreground from video sequences is a difficult problem in the presence of a rapidly changing background. In this paper, a novel technique for segmenting the foreground from video sequences with a dynamic background has been developed by treating the segmentation problem as a problem of classifying the foreground and background pixels of a video frame. For the purpose of this classification, a novel feature image has been constructed and used in the framework of a support vector machine. The feature image has been constructed by using the individual features representing the gray levels, hue and saturation levels of the image pixels. An attribute of the feature image leading to the computational simplicity of the proposed segmentation technique lies in its ability to represent multiple features of a pixel with a scalar value. Another distinguishing characteristic of the proposed method is that, unlike some other data classification based approaches for segmentation in which *a priori* knowledge of the object's shape and size is required or a set of training samples needs to be manually selected, the training samples employed by the classifier are automatically selected. The proposed method has been applied to a number of video sequences with a dynamic background to segment the moving foreground, and the results have been compared with those obtained by using some of the other existing schemes in the literature. In terms of the subjective and objective results for the segmentation, the proposed method has been shown to outperform the existing ones with an increased computational cost.

V. ACKNOWLEDGEMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the Regroupement Stratégique en Microélectronique du Québec (ReSMiQ).

The first author of the paper is now with Segway Robotics Inc.

REFERENCES

- [1] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Modules for Real-time Tracking", Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit., Vol.2, Jun. 1999, pp. 246-252.
- [2] O. Barnich and M. V. Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences". IEEE Trans. Image Processing, Vol. 20, No. 6, 2011, pp. 1709-1724.
- [3] M. Heikkilä and M. Pietikäinen, "A Texture-based Method for Modeling the Background and Detecting Moving Objects", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 28, No. 4, 2006, pp. 657-662.
- [4] B. Zhang, Y. Gao, S. Zhao and B. Zhong, "Kernel Similarity Modeling of Texture Pattern Flow for Motion Detection in Complex Background", IEEE Trans. Circuits Syst. Video Technol., Vol. 21, No. 1, 2011, pp. 29-38.
- [5] S.-C. Huang, "An Advanced Motion Detection Algorithm with Video Quality Analysis for Video Surveillance Systems", IEEE Trans. Circuits Syst. Video Technol., Vol. 21, No. 1, 2011, Jan.2011, pp. 1-14.
- [6] J. Zhang and C. H. Chen, "Moving Objects Detection and Segmentation in Dynamic Video Backgrounds", Proc. IEEE Conf. Technol. Homeland Security, May 2007, pp. 64-69.
- [7] J. Zhou, D. Gao and D. Zhang, "Moving Vehicle Detection for Automatic Traffic Monitoring", IEEE Trans. Vehicular Technol., Vol. 56, 2007, pp. 51-59.
- [8] D. Vernon. (1999). Machine Vision. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/BOOKS/VERNON/vernon.htm> [Oct. 11, 2012].
- [9] C. Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, Vol.20, Mar. 1995, pp. 273-297.
- [10] D. Decoste and B. Schölkopf, "Training Invariant Support Vector Machines", Machine Learning, Vol.46, Mar. 2001, pp.161-190.
- [11] A. Barla, F. Odone, and A. Verri, "Histogram Intersection Kernel for Image Classification", Proc. IEEE Int. Conf. Image Process., Vol. 3, Sept. 2003, pp. 513-516.
- [12] F. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces", Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit., Vol. 1, Jun. 2005, pp. 829-836.
- [13] Water [Online]. Available: <http://www.cs.bu.edu/groups/ivc/data/DynamicBackgrounds/ICCV2003/water/>
- [14] WaterSurface, Curtain and Trees [Online]. Available: <http://perception.i2r.a-star.edu.sg/>
- [15] J.Black, T. Ellis and P. Rosin, "A Novel Method for Video Tracking Performance Evaluation", Proc. Int. Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Oct. 2003, pp. 125-132.