2012 IEEE International Conference on Multimedia and Expo Workshops

# MOTION SEGMENTATION BASED ON 3D HISTOGRAM AND TEMPORAL MODE SELECTION

*Dibyendu Mukherjee and Q. M. Jonathan Wu*

Department of Electrical and Computer Engineering,
University of Windsor,
Ontario, Canada

## ABSTRACT

Motion segmentation has been a well explored research topic due to its vast application area. This work proposes a real-time motion segmentation method based on 3D histogram and temporal mode selection. The temporal distribution of a video sequence consists of the motion in the foreground and the relatively immobile background. A 3D histogram provides a short-term memory of the aforementioned distribution. The temporal mode selection process involves identifying the most frequent values in the distribution and construct the background thereafter. This work provides a detailed analysis of the proposed method along with an easy-to-implement algorithm. A number of experimental results and comparisons with some of the leading algorithms are provided to show that the proposed method can provide real-time, robust and highly accurate results.

***Index Terms***— Motion Detection, Real-time, Histogram

## 1. INTRODUCTION

Segmentation of motion from a video sequence has been a popular domain for research in computer vision. It has a vast application area consisting of traffic monitoring, surveillance, human-computer interaction etc. There have been several types of approaches used for motion detection. Among those, the most popular is background subtraction which consists of determining the background model, and subtract the background from each frame to extract the foreground information. Supposedly, foreground is fast changing and significantly different from the background. Thus, a simple subtraction should yield satisfactory results. But, in practice, there are several challenges that are increase the difficulty in the process. Some of the main problems are - slow foreground, inadequate background and nonstationary background. Another major factor for real-time operation is the computational complexity. Unfortunately, there exists an inverse relationship between the computational speed and the accuracy of the algorithms making it difficult to find an optimal solution.

Researchers have provided methods to counter the inherent problems and improve the results [1, 2, 3, 4, 5, 6]. Among the others, two approaches are worth to mention separately because of their simplicity and popularity - Gaussian Mixture Model (GMM) based approach and Sigma-Delta background estimation based approach. Stauffer and Grimson [7] have successfully applied probabilistic mixture model for background estimation. The process is real-time and moderately accurate. The work has been used as a model for many probability model based approaches [4, 5, 8, 9, 10, 11] thus claiming an undeniable position in the research. However, the approach suffered from the problem of manual assignment of number of clusters, slow adaptivity and accuracy issues. The second approach mentioned earlier is Sigma-delta based background estimation [1]. It is a simple method for motion detection. The key features of the approach are ease of implementation and fast computation. However, the approach fails for slow foreground and inadequate or nonstationary backgrounds. Some of the issues have been addressed by [11] which proposes a Conditional Random Field (CRF) based background/foreground/shadow segmentation. The method is reasonably accurate but lacks real-time computational speed. There are a number of other good approaches worth mentioning. But, due to the short span of the paper, we proceed to the description of the proposed approach.

In this work, we propose a very simple approach of background estimation. The method is based on a 3D-histogram that changes incrementally by a temporal block movement. The process is very fast and easy to implement. 3D histograms have been previously used [12]. But, the effectiveness of the proposed algorithm lies in its absence of preprocessing steps, minimal postprocessing steps and surprisingly low number of parameters used. Experimental results and comparisons with GMM, CRF and Sigma-delta are provided along with a pseudo code to verify the effectiveness of the algorithm.

The rest of the paper is divided as follows: The proposed approach has been discussed in detail in Section 2. Experimental results and comparisons are provided in Section 3. Finally, we conclude with some suggestion for future extensions in Section 4.

978-0-7695-4729-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMEW.2012.90

## 2. PROPOSED METHOD

A video sequence has three dimensions: rows and columns of an image frame (assuming all the frames are of same size) and the time axis. A pixel process is defined as the set of values taken by a particular pixel over time. At time instant $t$, for a pixel at position $(x, y)$, the pixel process can be expressed mathematically as

$$\{X_1, ..., X_t\} \text{ with } X_i = I(x, y, i), \tag{1}$$

where, $I(x, y, i)$ denotes the $D$-dimensional pixel intensity (gray scale or color) at position $(x, y)$ and time $i \in [1, t]$. It is quite reasonable to assume that over a long period of time, the most probable value that a pixel can take would belong to the background. This can be proved with an argument that a foreground object would stay at a pixel location for a finite and supposedly short amount of time. If the amount of time spent at a single pixel exceeds a certain threshold $T$ (which can arguably be taken as 30-60 frames because, a real-time video sequence contains a rate of 20-33 fps and a assumption of a foreground staying at a single pixel for 1-2 second is a very relaxed threshold), it can be assumed to be a part of background thus validating the assumption of most probable intensity value for a pixel. The cases of a very large or very slow moving objects are also considered when a large value of 50-60 is used for $T$. Consider a large moving object and a frame of $100 \times 100$ pixels. If the object covers a pixel for 60 frames, that means the object covers its own length in 60 frames. Thus, if the object is 100 pixels long (equal to frame length), then its speed is 1.67 pixels per frame, which is very low. But, the case is extremely exaggerated and is impossible in practical operation because, an object with a size equal to image size would reside very close to the image acquisition system itself. Thus, the object motion would be fast compared to the image size and would be much faster than 1.67 pixels per frame. Similarly, consider a small object of 5-10 pixels long. If the object stays at a pixel for 60 frames, that implies the object covers 0.08-0.16 pixels per frame and would take 600 frames to cover a $100 \times 100$ image. That means more than 18 seconds!

With that assumption taken so forth, it is well worth concluding that, over a fixed period of time, the background intensity $B$ at a pixel $(x, y)$ is the most-occurred intensity value at that pixel, or, in statistical terms, the *mode* of the pixel process

$$bg = \text{mode}(\{X_1, ..., X_t\}). \tag{2}$$

It would have been easy to compute the highest occurring intensity at a pixel and term it as background. That would obviously be correct and the background would be modeled easily. But, if the background changes for a pixel after a certain time, the new background would be continually detected as foreground till the occurrence number of the new background intensity exceeds that of the old background. As time progresses, this condition would be continually harder to satisfy. Thus, the time amount to consider the mode must be limited yet reasonable. Thus, in this work, a 3D-histogram is maintained. A 3D-histogram consists of the number of rows and columns versus the intensity frequency at the corresponding location over a finite time interval $T$. In this work, $T$ has been kept fixed at 20. The pseudo code for the algorithm is provided in Algorithm 1.

**Algorithm 1** Proposed algorithm

**procedure** MDETECT($vSeq, fSize, nF, T$)
  $hist3D \leftarrow zeros(fSize, 256)$
  $xBlock \leftarrow zeros(fSize, T)$
  **for** $i \leftarrow 1, nF$ **do**
    $x \leftarrow vSeq(i)$
    **if** $i \leq T$ **then**
      $xBlock(:, i) \leftarrow x$
      **for** $j \leftarrow 1, 256$ **do**
        $indx \leftarrow x$ AND $(j - 1)$
        $hist3D(indx, i) \leftarrow hist3D(indx, i) + 1$
      **end for**
    **else**
      **for** $j \leftarrow 1, 256$ **do**
        $indx \leftarrow x$ AND $(j - 1)$
        $hist3D(indx, i) \leftarrow hist3D(indx, i) + 1$
        $indx \leftarrow xBlock(:, 1)$ AND $(j - 1)$
        $hist3D(indx, i) \leftarrow hist3D(indx, i) - 1$
      **end for**
      **for** $j \leftarrow 1, T - 1$ **do**
        $xBlock(:, j) \leftarrow xBlock(:, j + 1)$
      **end for**
      $xBlock(:, T) \leftarrow x$
    **end if**
    $bg \leftarrow \arg\max(hist3D, 3)$
  **end for**
  **return** $bg$
**end procedure**

The algorithm takes as input, $vSeq$: the video sequence, $fSize$: size of each frame in $vSeq$, $nF$: number of frames in $vSeq$ and $T$. $hist3D$ and $xBlock$ represent the 3D histogram matrix and the temporal block of $T$ frames, respectively. These variables are initialized with zero matrices. The algorithm initially runs for $T$ frames, storing the contents of the frames in $xBlock$ and incrementing the intensity frequency value for each pixel. For that purpose, $indx$ stores the logical pixel locations corresponding to the gray values. After $T$ frames, for each new frame, the corresponding pixel intensity frequency is incremented in $hist3D$. Since, $xBlock$ contains the last $T$ frames, the previous $T^{\text{th}}$ frame would be out of scope in the current instance. Thus, the intensity frequencies are decremented for that corresponding frame in $hist3D$. This way, a running $T$ frame histogram is maintained in $hist3D$. Next, $xBlock$ is updated with current frame and the previous $T^{\text{th}}$ frame is removed. As already discussed, back-

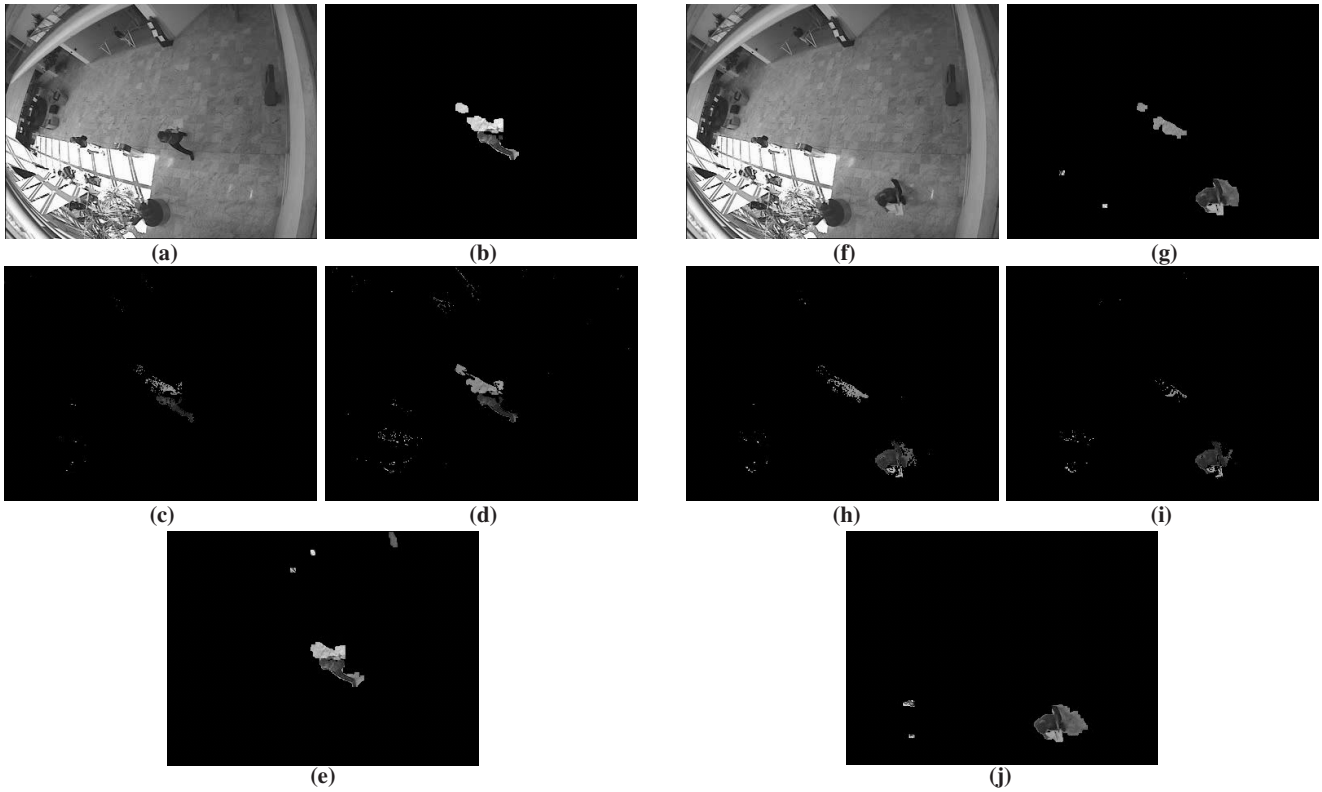


**Fig. 1**. First experiment: (a) Original video frame 33, corresponding (b) GMM output, (c) CRF output, (d) Sigma-delta output, (e) proposed output, (f) Output video frame 64, corresponding (g) GMM output, (h) CRF output, (i) Sigma-delta output, (j) proposed output

ground pixel values are the modes of the histogram for each pixel. Thus, finally, the background is obtained by getting the intensity values for each pixel, for which the frequency ($3^{\text{rd}}$ dimension of $hist3D$) is maximum.

The algorithm is easy to implement. Except for the temporal block length $T$, there are no parameters controlling the algorithm. $T$ can be reasonably fixed at a value between 20 to 33. Thus, the algorithm can run without manual intervention. Also, foreground can always be obtained using background subtraction. Both background and foreground obtained by the algorithm are demonstrated in the experimental section. Finally, the algorithm, although uses grayscale histogram, can also be used for color videos by converting them to grayscale. This does not reduce or change the performance of the algorithm, because, it does not use the color information.

## 3. EXPERIMENTAL RESULTS

The algorithm has been tested on CAVIER datasets and different types of video sequences ranging from low foreground contents to high amount of foreground contents. It has been compared with conventional GMM, Sigma-delta based method and CRF. Simple erosion and dilation with a fixed size binary element (disk of radius 3 pixels) has been performed on the foreground to remove certain subtle noises for all methods that are compared. Every algorithm has been coded in MATLAB and runs on a computer with AMD Phenom II X6 Processor with frequency of 3 GHz. For GMM and CRF, number of clusters has been kept constant at 3. An increase in this number can improve accuracy for some cases, but with a decrease in computational speed.

Current section has been divided into four sections. Section 3.1 demonstrates a situation of slow foreground that keeps a "ghost" image on the background. In Section 3.2, a busy road sequence with congested foreground is depicted with a camera from a slanted angle, thus hiding the original background most of the times. Section 3.3 shows the background extracted from the "Hall" sequence having radial motion. Radial motion is surprisingly tough for motion segmentation because, the moving object occupies a portion of the same set of pixels in the image frame while the size of the object changes, thus tending to register that portion of foreground as background. Finally, the fourth experiment in Section 3.4 provides a straight-forward comparison of all the algorithms based on the computational speed measured in terms of frame rates.

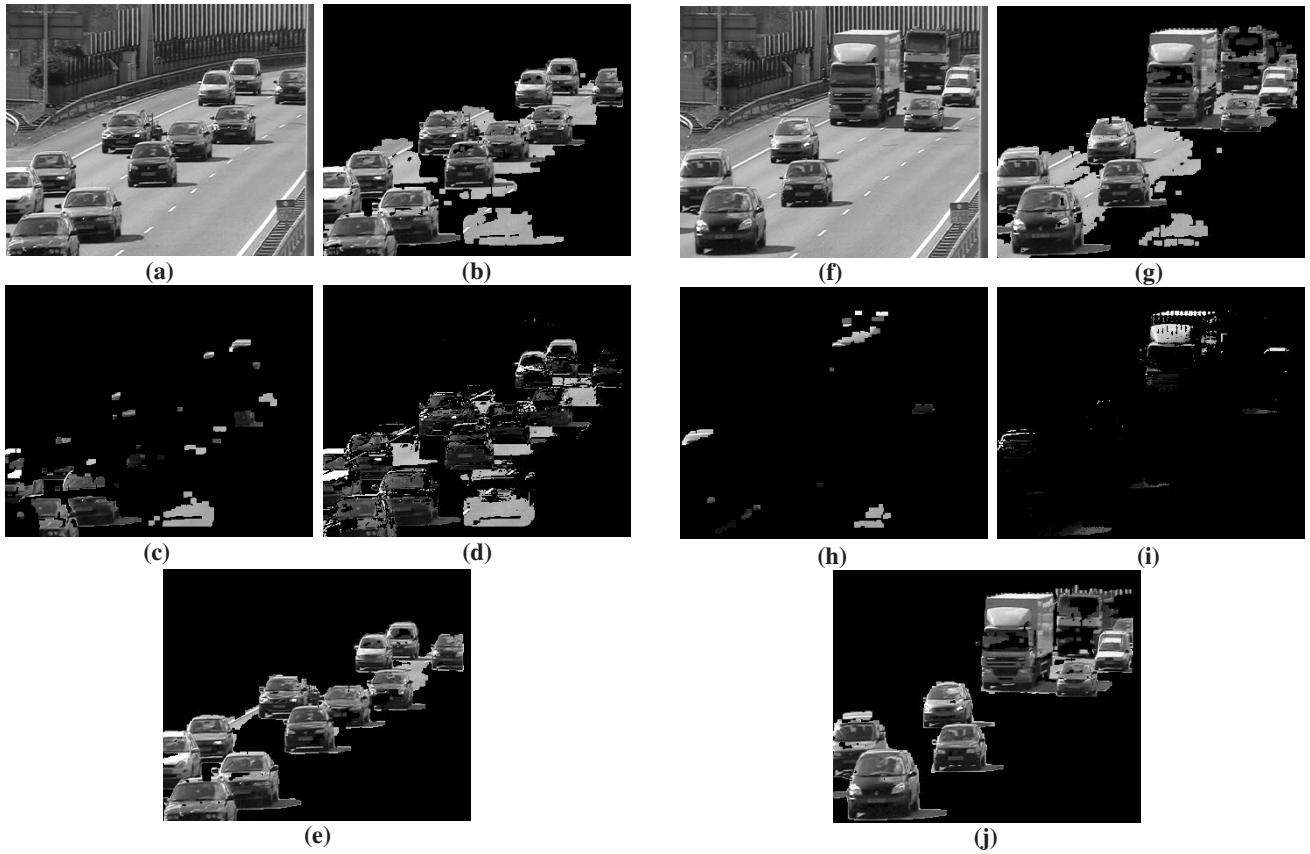


**Fig. 2**. Second experiment: (a) Original video frame 60, corresponding (b) GMM output, (c) CRF output, (d) Sigma-delta output, (e) proposed output, (f) Output video frame 170, corresponding (g) GMM output, (h) CRF output, (i) Sigma-delta output, (j) proposed output

### 3.1. Slow Foreground

For the first experiment, we show two frames from "Fight_RunAway" sequence (size $384 \times 288$ pixels) in Fig. 1. Fig. 1(a) shows frame 33 consisting of a man starting to move from a standing position while carrying a board in his hand. When he was standing still, he was assimilated in the background. As he starts to move, then motion is detected by each algorithm, as well as a "ghost" of his previous position because of the background subtraction. This ghost is detected by all of the algorithm with different accuracy.

In the frame shown in Fig. 1(f) (frame 64), the man has moved considerably. The results of segmentation are shown for each algorithm. The moving person is well detected by all the algorithms. But, a ghost remains for GMM, CRF and Sigma-delta. The proposed algorithm quickly recovers from the changes in the background, and thus the ghost disappears.

### 3.2. Congested Foreground

A video of busy road forms a very challenging sequence due to the continuous and congested motion of varying speed, as well as the unavailability of the background and changes in the illumination. If the background is unavailable, probability based models face the trouble of missing information and thus, sometimes fail to get the correct background. But, road sequences are the most common in surveillance where, a motion segmentation is used. Some sample frames and corresponding outputs from such a sequence are presented in Fig. 2. Fig. 2(a) and Fig. 2(f) show the frames 60 and 170 from the sequence, respectively. As can be seen from the results, GMM has performed somewhat well in Fig. 2(b) and Fig. 2(g), but still has mistakenly detected background road as foreground. CRF, on the other hand, has failed to detect the congested sequences in Fig. 2(c) and Fig. 2(h). The independent nature of the neighboring pixels from different moving objects and background do not help in the formation of the model for CRF. Also, the adaptivity of the CRF to the changing nature of the sequence is slower. Sigma-delta has performed better than CRF and GMM, but the output is severly affected by the increasing variance of the ever-changing foreground. Thus, this affects the detection more as the sequence progresses. This is depicted in Fig. 2(d) and Fig. 2(i). Fi-

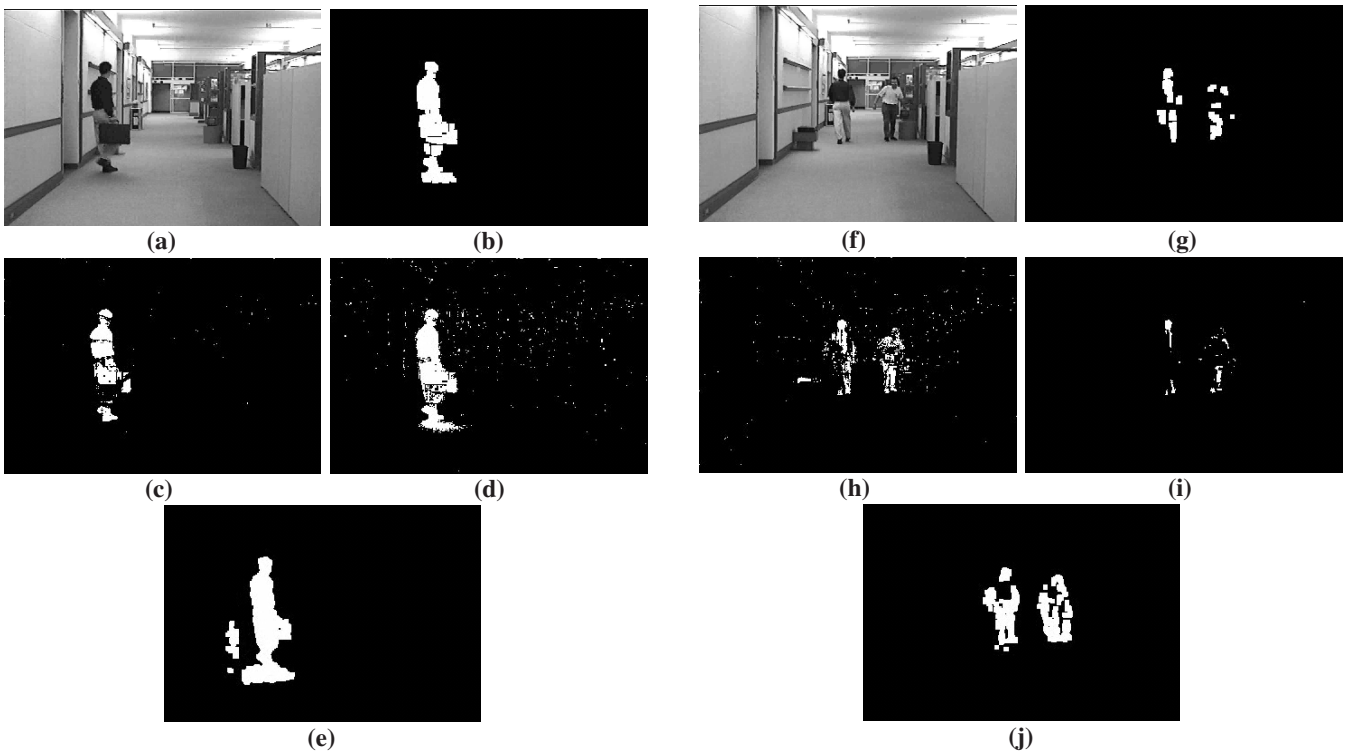


**Fig. 3**. Third experiment: (a) Original video frame 34, corresponding (b) GMM output, (c) CRF output, (d) Sigma-delta output, (e) proposed output, (f) Output video frame 170, corresponding (g) GMM output, (h) CRF output, (i) Sigma-delta output, (j) proposed output

nally, the proposed method outperforms the others with close to accurate detection in Fig. 2(e) and Fig. 2(j).

### 3.3. Radial Motion

The third experiment consists of background extraction in radial motion. A radial motion keeps the moving object occupying a part of the same pixels in a video sequence, thus creating a wrong belief of the object to be a part of the background. In Fig. 3, frame 34 and frame 170 from the Hall sequence (size $352 \times 240$ pixels) are shown with the corresponding foreground detection by different algorithms. The segmented foreground is not shown because, the man is wearing a black shirt which may be a camouflage with respect to the black background. In frame 34 shown in Fig. 3(a), the man is moving parallel to the image plane and thus, the background is well detected by GMM, CRF and Sigma-delta. The proposed algorithm also provides comparable output for this frame.

The segmented result drastically drops in performance for frame 160 in Fig. 3(f), where the man on the left moves radially after staying at a place for some time. GMM fails to provide proper detection in Fig. 3(g). CRF, on the other hand, being slow in adapting to the foreground that stays in same position for some time, provides a comparatively better result. It should also be kept in mind that the neighborhood relational information can be well used in these deterministic indoor sequences with relatively slower and limited motion. Sigma-delta, being a fast learner to the changing background, assimilates the slow-changing foreground in the background resulting in incorrect segmentation. The proposed method keeps a relatively large history of background. Thus, although the output is not perfect, it is comparatively better as shown in Fig. 3(j).

### 3.4. Computational Speed

In this section, the computational speeds of the different algorithms are compared. It is important to mention here, that all the algorithms have been implemented in MATLAB. Thus, the speed can be improved if the algorithms are implemented in a language like C or C++. We have used the "viptraffic" sequence (from MATLAB sample videos with size $160 \times 120$ pixels) of 120 frames and run the algorithms in the computer with the configuration mentioned earlier. Table 1 lists the algorithms compared with their total time taken and frames per second. The variations in frame rate for GMM and CRF are due to the amount of foreground present in a frame. The fractional frame rate for Sigma-delta and proposed approach are provided to show that the frame rates of these methods do

**Table 1**. Comparison of Computational speed

| Algorithm | Total time for 120 frames (seconds) | Frames per second |
|---|---|---|
| GMM | 6.14 | 19-20 |
| CRF | 41.20 | 2-4 |
| Sigma-delta | 1.19 | 100.84 |
| Proposed | 6.97 | 17.21 |

not vary with the amount of foreground. The Sigma-delta based method is super-fast compared to the other methods. Other than that, GMM and proposed method are comparable in speed while CRF is much slower in performance.

Based on the comparison of computational speed and the other experimental results, it can be observed that the proposed method provides highly accurate results with a speed comparable to other real-time methods. The speed is much slower compared to Sigma-delta, but the speed is tolerable for a real-time application. The method also has a very simple implementation procedure that does not depend on too many parameter initializations. Thus, it can be concluded that the method is an optimal choice for motion segmentation.

## 4. CONCLUSION

In this work, we have provided a simple method for motion segmentation based on 3D histogram and temporal mode selection. The approach is easy to implement, avoids large number of parameter initializations like other state-of-the-arts algorithms, and provides high quality segmentation results. The method is novel to our knowledge. The future work would include a search to find an automated choice of the temporal threshold parameter.


### ACKNOWLEDGEMENTS

This research has been supported in part by the Canada Research Chair Program, AUTO21 NCE, and the NSERC Discovery grant. The authors would also like to acknowledge the EC Funded CAVIAR project/IST 2001 37540 (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/) for the CAVIER datasets.


## 5. REFERENCES


[1] S. Toral, M. Vargas, F. Barrero, and M. G. Ortega, "Improved sigma-delta background estimation for vehicle detection," *Electronics Letters*, vol. 45, no. 1, pp. 32 –34, 2009.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Who? when? where? what? a real time system for detecting and tracking people," in *Int. Conf. on Auto. Face and Gesture Recog.*, 1998, pp. 222–227.

[3] S-C. S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *Jnl. Appl. Sig. Proc.*, vol. 2005, pp. 2330–2340, 2005.

[4] H. Greenspan, J. Goldberger, and A. Mayer, "Probabilistic space-time video modeling via piecewise gmm," *IEEE Trans. on Pat. Anal. and Mach. Intel.*, vol. 26, pp. 384–396, 2004.

[5] Y. Zhang, Z. Liang, Z. Hou, H. Wang, and M. Tan, "An adaptive mixture gaussian background model with online background reconstruction and adjustable foreground mergence time for motion segmentation," in *Int. Conf. on Indus. Tech.*, 2005, pp. 23–27.

[6] M. Harville, G. G. Gordon, and J. Woodfill, "Foreground segmentation using adaptive mixture models in color and depth," in *IEEE Workshop on Detect. and Recog. of Evnts. in Vid.*, 2001, pp. 3–11.

[7] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Trans. on Pat. Anal. and Mach. Intel.*, vol. 2, pp. 246–252, 1999.

[8] I. Schiller and R. Koch, "Improved video segmentation by adaptive combination of depth keying and mixture-of-gaussians," in *Scandinavian conf. on Img. anal.*, 2011, pp. 59–68.

[9] Q. Zhu, Y. Xie, J. Gu, and L. Wang, "A new video object segmentation algorithm by fusion of spatio-temporal information based on gmm learning," *Adv. in Auto. and Robo.*, vol. 123, pp. 641–650.

[10] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Trans. on Pat. Anal. and Mach. Intel.*, vol. 27, no. 5, pp. 827–832, 2005.

[11] Y. Wang, "Foreground and shadow detection based on conditional random field," in *Int. Conf. on Comp. Anal. of Img. and Pat.*, pp. 85–92.

[12] Wei Zhang, Q.M. Jonathan Wu, and Hai bing Yin, "Moving vehicles detection based on adaptive motion histogram," *Digi. Sig. Proc.*, vol. 20, no. 3, pp. 793 – 805, 2010.