A Robust Algorithm for Foreground Extraction in Crowded Scenes

Aravinda S. Rao, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami
Department of Electrical and Electronic Engineering, The University of Melbourne,
Parkville Campus, VIC - 3010, Australia.
Email: aravinda@student.unimelb.edu.au, {jgl, slaven, palani}@unimelb.edu.au

Abstract—**The widespread availability of surveillance cameras and digital technology has improved video based security measures in public places. Surveillance systems have been assisting officials both in civil and military applications. It is helping to identify unlawful activities by means of uninterrupted transmission of surveillance videos. By this, the system is adding extraneous onus on to the already existing workload of security officers. Instead, if the surveillance system is intelligent and efficient enough to identify the events of interest and alert the officers, it alleviates the burden of continuous monitoring. In other words, our existing surveillance systems are lacking to identify the objects that are dissimilar in shape, size, and color especially in identifying human beings (nonrigid motions). Global illumination changes, frequent occurrences of shadows, insufficient lighting conditions, unique properties of slow and fast moving objects, unforeseen appearance of objects and its behavior, availability of system memory, etc., may be ascribed to the limitations of existing systems. In this paper, we present a filtering technique to extract foreground information, which uses RGB component and chrominance channels to neutralize the effects of nonuniform illumination, remove shadows, and detect both slow-moving and distant objects.**

Index Terms—**Background segmentation, foreground extraction, frame differencing, occlusion, human tracking.**

I. Introduction

With the advancement in processing and computation technologies, surveillance systems have gained widespread attention due to their indispensable roles in our lives. Surveillance systems supply adequate information to policing authorities for further legal investigations, essentially protecting our community for better prosperity. However, most of the existing systems require manual intervention to monitor continuously and to identify the human activities, which is a tiresome task. Although we have our advanced modern-day engineering solutions and computational techniques in place, the systems are still devoid of sophisticated techniques to autonomously detect human activities.

In the process of motion detection and estimation for object tracking, background segmentation is the most critical of all. Often, the segmentation process can be normal, over-segmented or under-segmented [1]. The two most widely available methods for background segmentation are frame differencing [2] and background subtraction based on background modeling. The resulting difference can be binarized based on global thresholding [3] or multiple thresholds [4]. Additionally, accumulative difference image is an another way to segment the background from the foreground [5]. The most popular background modeling method is the weighted Mixture of Gaussians (MoG) model [6], [7].

Our objective of this work is to effectively extract foreground objects that are elastic in nature such as humans. We consider the background to be dynamic (constantly changing, occluded by objects and at times background may never be visible). We make no explicit assumptions of the object features (color, region, spatial connectedness, velocity etc). We also consider nonuniform illumination conditions, cast shadows and transient noise as inherently random and are part of the detection process. In addition, we would like the detection to be computed in near real-time (without buffering more than one frame for computation and having a previous output) and the output be binarized so that the logic 1 values correspond to the foreground.

In this paper, we present a new processing technique that is purely based on consecutive frame differencing. The technique uses RGB component channels, converts RGB to grayscale and chrominance signals , handling multi-object cases efficaciously. The filtering technique is shown to provide comparable accurate foreground extraction. The technique was applied on three different datasets and the results have been presented along with execution time for individual datasets. We are not comparing our execution times with others, but are mentioning the time taken by our technique for three different datasets close to practicable real-time processing.

This paper is organized as follows: Literature review is presented in Section II. Section III provides the details of the technique followed by the results in Section IV. Discussions and conclusion are included in Sections V and VI, respectively.

II. Literature Review

In order to extract foreground information from the scene, we would ideally like to build a model of the scene from the instant the view is available. One way is to estimate the background from the scene and subtract from the new frame. The other way is to extract foreground directly such as intensity thresholding. Kilger [3] used intensity-based thresholding to separate the fast-moving and slow-moving/no-motion objects from the background. However, determination of optimal threshold remained empirical. Kaup and Aach [8] proposed a novel algorithm to predict the uncovered background using spatial correlation and motion information along with temporal correlation. However, the ever changing dynamic background would not be able to uncover the background using spatial correlation nor temporal history.

In 1997, Wren *et al.* [9] used a single-Gaussian multiclass statistical pixel-based model (Pfinder) for tracking and detection of people. The drawback was that the scene was modeled assuming relatively static background with a single foreground object to be tracked. Stauffer and Grimson [6], [7] generalized the notion of adaptive background modeling using mixture of Gaussians (MoG). In this approach, pixel values were not explicitly classified as belonging to a single distribution, instead, values were based on mixture of Gaussians. Each pixel value was modeled as a mixture of Gaussian of recently observed pixel values. A new Gaussian was created by replacing the least probable Gaussian from an open-ended list of Gaussians to incorporate the new pixel value. Depending on the consistency and variance of the pixel values, the pixel that did not fit into any of the Gaussians was labeled as foreground. Elgammal *et al.* [10], [11] modeled the background using kernel density estimation of recent values of a pixel. In general, the Gaussian-based background estimations perform well when the background covers most of the area of the scene, is less fluctuating and the objects in the scene are mostly inelastic. Algorithms applied to rigid objects are relatively easier because of the spatial connectedness of the objects.

Oliver *et al.* [12] used observed background variations for eigenspace creation and followed by eigenvalue decomposition for background subtraction. However, we would like to have no prior knowledge of the scene. The correlation property of the background used by Seki *et al.* [13] holds only when the background is varying based on certain mathematical model and area to be filled is small in comparison to the objects that created the patch. Algorithms such as running running Gaussian average [14] lose edge information, which is necessary in identification of occluding contours [15].

Modeling of the background based on minimum and maximum variation of the pixels [16] would suit for backgrounds where change is relatively stationary. Techniques such as temporal median filter [17] requires large amounts of memory, produces delay in the output. Moreover, if the background is nonstationary (occupied with moving objects), background modeling would not capture the true background information.

Zhao *et al.* [18] used the approach of [9] to model background, where each pixel is modeled as an independent color Gaussian distribution. However, they use the background model based on the scene where no objects are present. In 2003, Zhao and Nevatia [19] approached the problem of segmentation (human crowd with occlusion) using 3D models to interpret the foreground, but shape models require definitive position, angle and size of the objects for accurate results. In 2011, Barnich and Droogenbroeck [20] presented a universal background subtraction technique based on Euclidean distance between the new pixel value and the existing values, but this method requires sufficient number of samples.

Although many methods exist for segmentation, each one of them produce accurate results under certain conditions. The idea of our work is to develop a generalized and quick approach for foreground extraction in monitoring crowded spaces. Due to the variation of lighting condition of these uncontrolled environment such as stadiums, tunnels, stations and other public spaces during the course of the day, there is a critical need to develop robust algorithms which extract foreground quickly enough to accommodate environmental changes. In doing so, we have selected 2 publicly available datasets and collected a unique dataset from Melbourne Cricket Ground (MCG) to demonstrate the effectiveness of the proposed technique. The following section provides the methodology used for background segmentation under different circumstances by controlling the sensitivity of the filter.

III. Methodology

This section provides the detailed implementation of the methodology. We are utilizing RGB and chrominance channels for extracting the foreground. The equations provided in this section were derived experimentally. To begin with, in a given frame, every pixel (for a given color space) has a value associated with it. Let us consider a pixel, $I(x_i, y_j)$, where I stands for image; x_i and y_j represent the coordinate of a two-dimensional space X and Y; i represents the i^{th} row and j represents the j^{th} column. The color space S_c contains different elements based on the color model (c) chosen. An image $I(x_i, y_j, t_n)$ refers to a frame at time t_n, n refers to the time sequence index (i.e. $t_n, t_{n+1}, \cdots, < \infty$) in a video sequence. Therefore, for any pixel $I(x_i, y_j, t_n)$ in a video frame, the following holds true

$$I(x_i, y_j, t_n) \subseteq S_c, \quad x_i, y_j \in \mathbb{Z}, \tag{1}$$

where $0 \leq S_c \leq L - 1$. Also, $L := \{l_i : 0 < l_i \leq K, \text{ and } l_i, K \in \mathbb{Z}_+\}$, where i is the indicator of the level, K defines upper-bound on L. The simplest way of detecting the motion is to take the difference of two frames given by:,

$$I(x_i, y_j)_{n+1} = I(x_i, y_j, t_{n+1}) - I(x_i, y_j, t_n) \tag{2}$$

To account for negative values, we take the absolute value of the difference, is given by:

$$|I(x_i, y_j)_{n+1}| = |I(x_i, y_j, t_{n+1}) - I(x_i, y_j, t_n)| \leq L \tag{3}$$

The binary classification of the objects are based on a threshold, given by:

$$I_B(x, y) = \begin{cases} 0, & \text{if } |I(x_i, y_j)| < l, \\ 1, & \text{if } |I(x_i, y_j)| \geq l \end{cases} \tag{4}$$

where $0 < l < L$, I_B stands for binary image. Usually, the threshold l is decided based on the modality distribution of $|I(x_i, y_j)|$.

The flowchart of the proposed approach is shown in the Fig. 1 and the steps are detailed below:

1) In order to provide complete change in the scene, the absolute values of successive frame differences with

threshold set at zero intensity is computed:

$$\begin{aligned} I(x,y)_{abs} &= |I(x_i,y_j)_{n+1}| > 0 \\ &= |I(x_i,y_j,t_{n+1}) - I(x_i,y_j,t_n)| > 0 \end{aligned} \tag{5}$$

2) Forward filtering using green and blue channels was performed (note that the operations from steps 2 to 5 are being performed on second frame i.e. I_{forward} represents $I(x,y,t_{n+1})_{\text{forward}}$):

$$I_{\text{forward}} = \left\lfloor I(x_i,y_j,t_{n+1},\text{g})^2 - I(x_i,y_j,t_{n+1},\text{b}) \right\rfloor \tag{6}$$

This filter essentially uncovers the small magnitude intensities and suppresses the existing predominant intensities.

3) Next, the inverse of the filter is provided by first replacing the null values of I_{forward} given by:

$$I(x_i,y_j,t_{n+1})_{\text{replace}} = \begin{cases} 1, & \text{if } I_{\text{forward}} = 0 \\ I_{\text{forward}}, & \text{otherwise} \end{cases} \tag{7}$$

Now, the inverse of the filtering operation was performed to enhance the naturally dominant light sources and is given by:

$$I(x_i,y_j,t_{n+1})_{\text{inverse}} = \frac{1}{\left| I(x_i,y_j,t_{n+1})_{\text{replace}} \right|} \tag{8}$$

4) Further, the difference between $I(x_i,y_j,t_{n+1},\text{g})$ and $I(x_i,y_j,t_{n+1},\text{b})$ would provide the essential objects with constantly illuminated regions. To remove the illumination effects, we use:

$$I_{\text{illumination}} = |I(x,y,t_{n+1},\text{g}) - I(x,y,t_{n+1},\text{b})| \tag{9}$$

5) Chrominance signals, which supply information about the objects that are sensitive to Cb and Cr channels were processed:

$$I_{\text{CbCr}} = (Cr \cup I_{\text{illumination}}) + Cb^{'} \tag{10}$$

where $Cb^{'}$ is the negative (complement) of Cb channel. This forms the sensitivity of the filter.

6) Finally, the foreground objects were obtained by performing the following operation:

$$\text{objects} = \left\{ \{ (I_{\text{forward}} - I_{\text{inverse}}) - I_{\text{CbCr}} \} + I(x,y)_{\text{abs}} \right\} \tag{11}$$

Note: the system variables *u* and *l* represent the intensity of percentage of upper and lower pixel values derived from histogram of the respective processed images to be saturated in order to enhance the objects; *k* is the sensitivity of the filter to extract the foreground effectively.

IV. Results

The above technique of processing the video was applied on three different datasets for evaluation. Fig. 2 depicts the output from various stages applied on Context Aware Vision using Image-based Active Recognition (CAVIAR) dataset [21]. Likewise, Fig. 3 and Fig. 4 delineate the outputs when applied on Advanced Video and Signal based Surveillance (AVSS) [22] and Melbourne Cricket Ground (MCG) datasets respectively. The captions of each of the figure contain the detailed sequential steps involved in processing. All video files were converted manually to Audio Video Interleave (AVI) format before the method was applied. WMV version 1 (WMP7) was used for conversion into AVI format. Table I provides the details of the dataset information. The frames were processed sequentially (with no skips between frames). All implementations were done in MATLAB 7.12 using Computer Vision System Toolbox on Windows XP (SP2 Professional, 32-bit system) equipped with an Intel® i7−2600 CPU running at 3.4 GHz. The system also included 512 MB ATI Radeon™ HD 5450 Graphics card. From the results, it is clear that the frame difference output alone is insufficient to provide information for future processing. To illustrate, Fig. 2a and Fig. 3a output provides meagre information in spite of absolute frame difference. In contrast, the variance is high in video of Fig. 4a, providing more information. Figs. 2b, 3b, and 4b denote the regions in a video sequence where the illumination is dominantly present. Figs. 2g, 3g and 4g portray the output after performing morphological operations. A disk radius of 3 was used for image closing and a 3×3 median filter was used for removal of speckles. Figs. 2h, 3h and 4h were the second frames, respectively, from the video sequence. The illustrated technique uses only the second frame for processing except for frame difference (step 1), where in both the second and first frame are used. Fig. 3g demonstrates the ability of the technique to detect abandoned object (stationary) as well as moving objects; shadow elimination is shown in Fig. 4g together with illumination normalization; sensitivity of the technique to detect slow-moving and distant objects, and occluded objects can be seen in Fig. 2g. The blue channel sensitivity, *k_cb*, was set to 0.35, 0.2 and 0.9, respectively, for CAVIAR, AVSS and MCG videos. The red channel sensitivity, *k_cr* was set to 0.9, 0.7 and 0.9 correspondingly. However, when the object is perceptually brighter, the method fails to

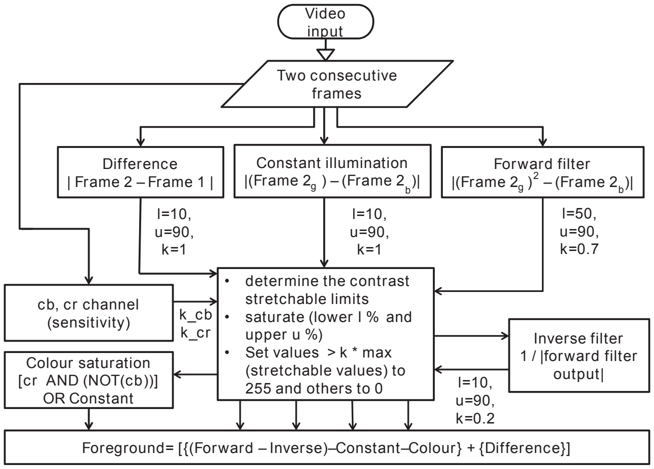
Video input
Two consecutive frames
Difference
| Frame 2 – Frame 1 |
Constant illumination
|(Frame 2g) – (Frame 2b)|
Forward filter
|(Frame 2g)2 – (Frame 2b)|
l=10, u=90, k=1
l=10, u=90, k=1
l=50, u=90, k=0.7
cb, cr channel (sensitivity)
k_cb
k_cr
• determine the contrast stretchable limits
• saturate (lower l % and upper u %)
• Set values > k * max (stretchable values) to 255 and others to 0
Inverse filter
1 / |forward filter output|
Colour saturation
[cr AND (NOT(cb))] OR Constant
l=10, u=90, k=0.2
Foreground= [{(Forward – Inverse)–Constant–Colour} + {Difference}]

Fig. 1. Flowchart of foreground extraction

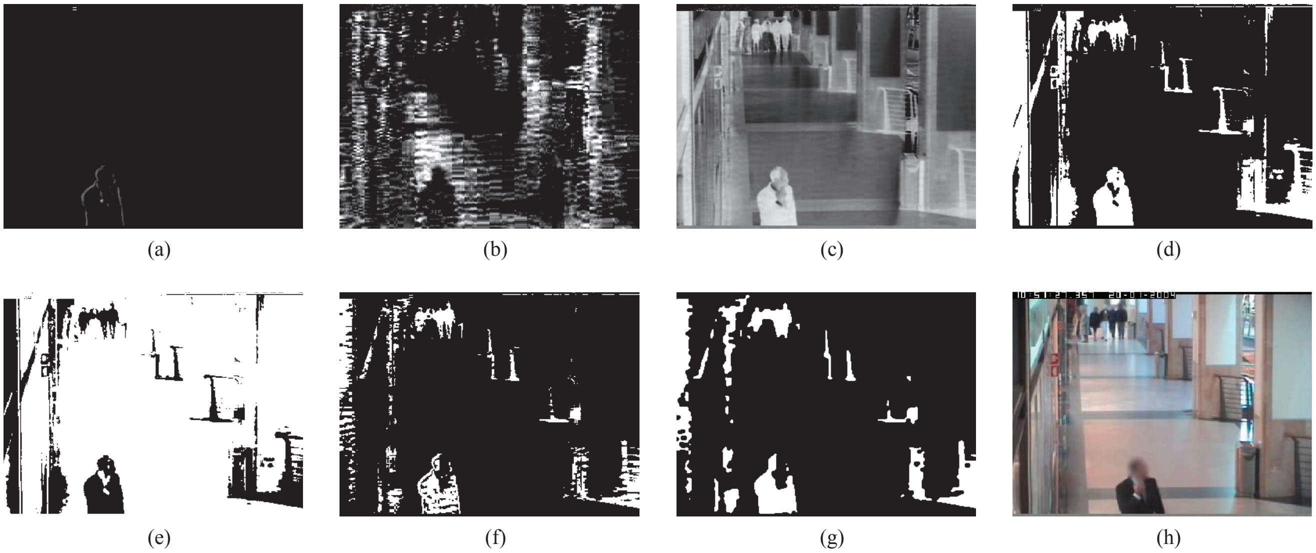

Fig. 2. Dataset from Context Aware Vision using Image-based Active Recognition (CAVIAR) [21]—(“WalkByShop1cor.mpg”). (2a) frame difference output $(= 220 - 219)$, (2b) constantly illuminated region $(G - B)$, (2c) illumination normalization $(G^2 - B)$, (2d) output of forward filter, (2e) output of inverse filter $(\frac{1}{G^2 - B})$, (2f) background segmented, (2g) background segmented after morphological operations (image closing with disk radius=3, 3×3 median filter), (2h) original input (frame 220).

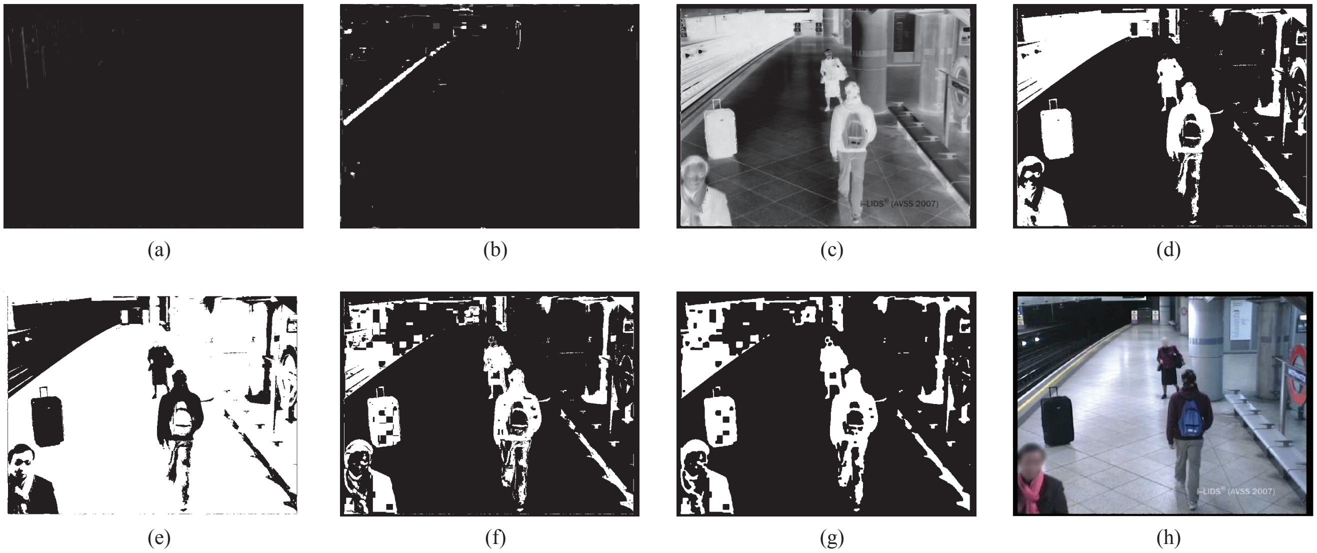

Fig. 3. Dataset from Advanced Video and Signal based Surveillance (AVSS) [22]—(AVSS AB Hard). (3a) frame difference output $(= 3649 - 3648)$, (3b) constantly illuminated region $(G - B)$, (3c) illumination normalization $(G^2 - B)$, (3d) output of forward filter, (3e) output of inverse filter $(\frac{1}{G^2 - B})$, (3f) background segmented, (3g) background segmented after morphological operations (image closing with disk radius=3, 3×3 median filter), (3h) original input (frame 3649).

detect stationary and slow-moving objects. Nevertheless, this can be addressed by using existing background subtraction and motion estimation algorithms. Table II provides the processing time taken by different datasets.

V. Discussion

For a given video frame $I(x_i, y_j)_n$, let $O := \{o : o \in I(x_i, y_j)_n\}$ be the pixels corresponding to the foreground objects and $B := \{b : b \in I(x_i, y_j)_n\}$ be the background pixels such that $\{O \cap B = \emptyset\}$. The major drawback of the global thresholding approach is that the distribution of pixels, which is determined statistically, are discarded from a collective set of pixels as background. More formally, let $D := \{d : 0 \neq d \in I(x, y)_i\}$, then the threshold operation may be considered as injective mapping of a subset of elements of D to the foreground F. i.e. let $W \subset D$, then,

$$T : W(T) \to F(T)$$

where $\forall w \in W > l$. However, T fails to consider the elements $D \cap W \neq \{\emptyset\}$, which would limit the foreground to $F|_W$

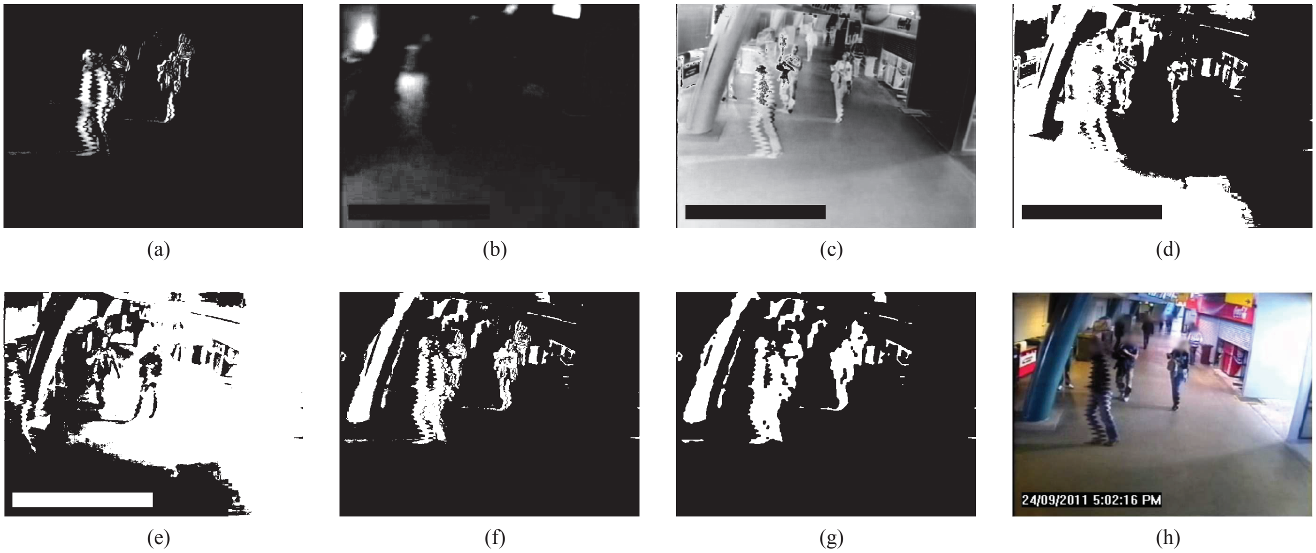

Fig. 4. Dataset from Melbourne Cricket Ground (MCG). (4a) frame difference output (frame $= 13 - 12$), (4b) constantly illuminated region $(G - B)$, (4c) illumination normalization $(G^2 - B)$, (4d) output of forward filter, (4e) output of inverse filter, (4f) background segmented, (4g) background segmented after morphological operations (image closing with disk radius=3, 3×3 median filter), (4h) original input (frame 13).

TABLE I
DATASET INFORMATION

Dataset		CAVIAR	AVSS	MCG
Original	Size (in pixels)	384×288	720×576	680×480
	Frame rate (per sec)	25	25	30
	Data rate (kbps)	1155	3911	1868
	Duration (minutes)	$1:34$	$3:38$	$14:01$
	File format	MPEG	AVI	ASF
Converted	Frame rate (per sec)	25	25	25
	Data rate (kbps)	829	3047	1502
	Duration (minutes)	$1:34$	$3:39$	$14:01$
	File format	AVI	AVI	AVI

TABLE II
TIME TAKEN TO PROCESS VIDEO FOR FIRST 100 FRAMES

Dataset	CAVIAR	AVSS	MCG
Initialization time (s) *	1.386380	1.4954050	1.4356340
Total time (s)	9.478341	19.884810	16.581148
Average processing time (s)	0.080919	0.1838940	0.1514551
For 30 frames (s)	2.427588	5.5168215	4.5436542

* initialization time is the time to produce the first iteration output

instead of $F|_D$. Then, considering the nonlinear operator T such that $T : W(T) \rightarrow F(T)$

$$F = \Big\{ T\Big(|I(x_i, y_j)_{n+1}| \leq L\Big) \cup T\Big(|I(x_i, y_j)_{n+1} = 0|\Big)\Big\} \tag{12}$$

where, the first term in the righthand side of the equation is controlled by the threshold level. Therefore, it is evident that the thresholding operation alone limits the available information to be transferred to the foreground set F.

The frame difference between two sequential video frames can segment the foreground from the background based on the applied threshold. However, the degree to which this can be achieved depends on the motion component of the scene at different regions and on the threshold. The higher the motion the more prominent the change is and consequently, more likely the change will be detected; hence, the thresholding can easily separate the background and foreground. To achieve this either the motion component has to be increased by skipping a few frames and then find the difference, or process the difference between consecutive frames using another method so as to obtain the motion component quite clearly. We adopted the latter approach in our methodology. The moving objects closer to camera are easily identified in case of background segmentation is by the fact that they exhibit distinguishable variance from the background. This can be reasoned out as a direct result of increased variance: the vertical and horizontal motion appear distinctly compared to the rest of the regions. The traits of slow-moving objects are closely coupled with that of background in terms of temporal features such as the displacement. Hence, to ascertain the detection of both slow-moving objects (even a countably infinitesimal change) and fast-moving objects, and distant objects, we adopted the absolute frame difference and the threshold was set to the smallest change by $(|l_i| > 0)$.

It is observed that when there is a saturation of light in a particular region, the three independent color channels tend to saturate equally. Because of this, most of the methods that extract foreground from the background are unable to give equal weightage to all the regions in a frame. By calculating $G^2 - B$ (as in step 2), we are in fact, focusing on unilluminated regions: saturated regions are negated in the process of subtraction and small intensity values are amplified. The regions of constant illumination are found by performing

$G - B$ result. The k_cb and k_cr values usually lie between 0.3 and 0.9. Increasing these values introduce noise when no objects are found in the scene. The increased values accentuate small object variations and consequently the image noise. The values should be kept low unless extremely high sensitivity is required. This method compliments the existing algorithms in that the combination of the above-mentioned method and the existing foreground extraction methods would greatly improve the segmentation of the background from foreground.

The elastic nature of human motion is handled by means of union of $G^2 - B$ and its inverse $\frac{1}{G^2-B}$ (zeros excluded by replacing them with 1) to include both the occluded and unoccluded objects. This method is sensitive to be operated under covered regions and highly darker objects. However, there will be circumstances where the method can fail to segment the foreground objects completely when the scene is entirely illuminated. In this case even the existing methods would be unable to recognize the motion. Nonetheless, efficient implementation of a tracking algorithm can determine the path of the objects and recover the parts of the objects to a greater extent. Depending on the scene, the sensitivity of the chrominance channels can control the degree of detection of both moving and stationary objects. In our methodology, we did not explicitly handle problems with shadows. Shadows were eliminated to a greater extent with help of filtering techniques presented above (implicit shadow removal by combination of different signal processing presented previously). Further analysis is required in handling shadows under different conditions.

VI. Conclusion

In conclusion, a foreground extraction technique using component, intensity and chrominance channels has been presented. The segmentation of the foreground from the background was achieved by taking the absolute frame differences and adjusting the sensitivity of a given frame by making use of component and chrominance channels. Component signals were used for reducing change in illumination and removal of shadows. Grayscale was used for motion detection based on frame differencing and absolute thresholding. Chrominance channels were used to set the sensitivity of segmentation process. The method also demonstrated to be useful in stationary object detection as well. The proposed method was tested on three different datasets with promising results.

Acknowledgment

This work is partially supported by the ARC linkage project LP1232434, partnering the University of Melbourne, Melbourne Cricket Club and ARUP. Authors thank Paul Stanley, Andrew Maher, George Kazantzidis (all from ARUP), Trevor Dohnt (from MCG) and the staff of MCG who were very helpful during the project.

References

[1] Z. Zhang, H. Gunes, and M. Piccardi, “Tracking people in crowds by a part matching approach,” in *Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, ser. AVSS ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 88–95.

[2] R. Gonzalez and R. Woods, *Digital Image Processing (3rd Edition)*. Prentice Hall, 2007.

[3] M. Kilger, “A shadow handler in a video-based real-time traffic monitoring system,” in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 1992, pp. 11–18.

[4] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[5] R. Jain and H. H. Nagel, “On the analysis of accumulative difference pictures from image sequences of real world scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 206–214, 1979.

[6] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, “Using adaptive tracking to classify and monitor activities in a site,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 22–29.

[7] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 246–252.

[8] A. Kaup and T. Aach, “Efficient prediction of uncovered background in interframe coding using spatial extrapolation,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, 1994, pp. V/501–V/504.

[9] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: real-time tracking of the human body,” in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 51–56.

[10] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Computer Vision ECCV 2000*, ser. Lecture Notes in Computer Science, D. Vernon, Ed. Springer Berlin / Heidelberg, 2000, vol. 1843, pp. 751–767.

[11] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[12] N. M. Oliver, B. Rosario, and A. P. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[13] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, “Background subtraction based on cooccurrence of image variations,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. II–65–II–72.

[14] J. Zhou and J. Hoang, “Real time robust human detection and tracking system,” in *Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition - Workshops*, 2005, pp. 149–149.

[15] B. K. P. Horn, *Robot vision*. Cambridge, MA, USA: MIT Press, 1986.

[16] I. Haritaoglu, D. Harwood, and L. S. Davis, “W^4: Who? when? where? what? a real time system for detecting and tracking people,” in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 222–227.

[17] I. Haritaoglu, D. Harwood, and L. S. Davis, “A fast background scene modeling and maintenance for outdoor surveillance,” in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 179–183 vol.4.

[18] T. Zhao, R. Nevatia, and F. Lv, “Segmentation and tracking of multiple humans in complex situations,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. II–194–II–201.

[19] T. Zhao and R. Nevatia, “Bayesian human segmentation in crowded situations,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. II–459–66.

[20] O. Barnich and M. Van Droogenbroeck, “Vibe: A universal background subtraction algorithm for video sequences,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[21] R. Fisher, “Clips from shopping center in portugal (2nd set),” 2003-2004. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

[22] “i-lids bag and vehicle detection challenge,” 2007. [Online]. Available: http://www.eecs.qmul.ac.uk/∼andrea/avss2007_d.html