Journal of Computational Information Systems 8: 2 (2012) 493–505
Available at http://www.Jofcis.com

A Qualitative and Quantitative Comparison of Real-time Background Subtraction Algorithms for Video Surveillance Applications ⋆

M. Hedayati, Wan Mimi Diyana Wan Zaki∗, Aini Hussain

Department of Electric, Electronics and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia

Abstract

Background subtraction is a widely used technique for segmenting a foreground object from its background. The aim of this paper is to review and compare the performance of the most common statistical background subtraction methods, including median-based, Gaussian-based and Kernel density-based approaches. To obtain a fair evaluation, four challenging scenarios were selected based on Wallflower datasets. All review methods are based on processing speed, memory usage and segmentation accuracy. The overall evaluation shows that the Gaussian-based method gives the best performance in accuracy, speed and memory consumption. In addition, this paper provides a better understanding of algorithm behaviours applied to different situations for real-time video surveillance applications.

Keywords: Background Subtraction; Real-time Video Surveillance; Gaussian Mixture Modal; Median; KDE

1 Introduction

Today, video surveillance systems that are used as remote eye security systems play an important role in maintaining security and safety in our society. Today, these tools can be seen everywhere, including in residential areas, junctions, malls and airports. Real time human detection and behaviour analysis of people in enclosed areas are the main challenges for any smart Video Surveillance system. Accurate segmentation of moving objects from their background is the principle operation in this type of system.

Object detection approaches have been introduced by researchers over the past three decades; undoubtedly, the Background Subtraction (BGS) method is the most widely used technique for video security applications because of its simplicity, its acceptable accuracy and its low computation time. Although most BGS algorithms work well under simple conditions, there are various challenges that affect the output of the system. These challenges can be categorised into two main

⋆Project supported by MOSTI and OUP-UKM-ICT-36-184/2010 eScience Fund grant 01-01-02-SF0563.

∗Corresponding author.
Email address: wmdiyana@eng.ukm.my.com (Wan Mimi Diyana Wan Zaki).

January 2012

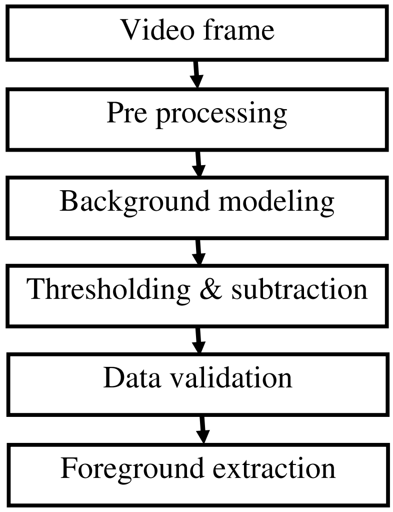


Fig. 1: General flow diagram of the BGS algorithm

groups: first, the system limitation and second, the environmental challenges. System limitations are directly related to the characteristics of the platform that is used, such as memory consumption and computational speed. Environmental challenges are imposed by the environment and are caused by light reflection, shadow, illumination changes, colour similarity or camouflage and background variation.

This paper comprehensively investigates the most popular background subtraction algorithms introduced by researchers, namely Median filtering, Approximate Median, Running Gaussian Average (RGA), Gaussian Mixture Modal (GMM) and Kernel Density Estimation (KDE). The goal is to achieve the best comparison of all of the algorithms evaluated with respect to accuracy, speed and memory consumption.

The rest of this paper is organised as follows: Section 2 reviews previous work on background subtraction and briefly explains thresholding and data validation techniques. Section 3 describes our evaluation methods and the features of each data set. Section 4 compares the selected methods and discusses the final results. Finally, Section 5 offers conclusions and provides recommendations for further work, based on the results presented.

2 Background Subtraction Algorithm Review

Briefly, to perform background subtraction, first the background has to model. The background is obtained, and modelled. Then, the incoming frame is obtained, and the background model is subtracted out. With the background model used in this way, a moving object can be detected. This algorithm is properly named background subtraction. The efficiency of a background subtraction technique correlates strongly with three important steps: modelling, thresholding and data validation (Figure 1).

Background modelling, as the backbone of the BGS algorithm, describes model representation and model adaption characteristics. Model representation defines the type of model selected to represent the background, and the representation can simply be a frame at time (t-1) or a statistics formula such as the median model. Model Adaption is the procedure used for adjusting the background changes that may occur in a scene. Thresholding, which is one of the basic image processing techniques, is a procedure that eliminates an unwanted range of pixels in the scene with respect to certain threshold values, whereas data validation is involved with the collection of techniques to reduce the misclassification of pixels.

Among the various methods introduced in the literature, we thoroughly review five distinct background modelling methods here, namely, Median, Approximate Median, Running Gaussian Average (RGA), Gaussian Mixture Modal (GMM) and Kernel Density Estimation (KDE). Based on their representation models, these methods can be divided into three categories, namely Median-based [2,3,4,12], Gaussian-based [22,23,24,25] and KDE-based [6,13,26] approaches. Moreover, brief reviews of thresholding and data validation techniques [1,5,17,20] are also discussed.

2.1 Background modelling

2.1.1 Median-based method

Median modelling is one of the simplest statistical background modelling methods, and because of its acceptable performance, it has reliably become a consideration for most researchers and developers. Median filtering is considered to be a recursive technique, which means that the algorithm stores the previous L video frames in a buffer and estimates the background based on variations in pixels in the buffer.

In [2,4], Cucchiara, R. et al. prove that, with a proper selection of the observation time window (nt), median filtering gives the best overall performance for real time applications as compared to mean and mod filtering, even with a limited length of sequence. There are two common approaches in median-based models, namely median filtering and approximate median filtering.

Median Filtering: In this model, the background is defined based on the median value of each pixel in all previous frames in the buffer. Thus, the background at time t can be defined as

$$B_t = U(I_t, I_{t-\Delta t}, ..., I_{t-(n-1)\Delta t}) \tag{1}$$

U is the updating model, where It is the frame at time t. In median filtering, the correct selections of the buffer size (n) and the frame time rate (t) are critical issues that affect the performance of the median filtering.

Cucchiara, R. et al [3] used median filtering on colour space to improve pixel selection. In [11], median and variance filtering are combined together to reduce false positive rates caused by the light reflection from trains in a train station. The complexity of the computing in the median filtering algorithm is equal to O (L log L).

Approximate median: As a complimentary method to median filtering, McFarlane, N. and Schofield, C. [12] introduce the approximate median method, which is a recursive technique. This model uses recursive filters to estimate median filtering so that the computational time and memory consumption of the developed system will be reduced [14,15]. The main drawback of this model is the adaptive rate, which requires a large amount of time to learn a new background as the background changes.

2.1.2 Gaussian-based method

The Gaussian technique is a statistical approach that models pixel intensities based on Gaussian probability distributions. Gaussian models have recursive patterns and can adaptively update the background without using a large buffer. There are two common approaches of Gaussian-based methods. The first approach is the Running Average Gaussian (RGA), which is the fastest algorithm among adaptive background subtraction methods and adapts the background using only one or two parameters [15]. The second Gaussian-based method is the Gaussian Mixture Model (GMM) [23,25].

Running Gaussian Average (RGA): This method uses a single Gaussian probability to model the colour distribution of the background and to employ basic adaptive filtering Eq. (2) to adopt changes in the scene (such as illumination changes) [25].

$$\mu_{t+1} = \alpha F_t + (1 - \alpha)\mu_t \tag{2}$$

where μ_t is the mean value at time t.F_t is the current frame and α is the updating rate. The formula in Eq. (2) was later updated by Koller, D. et al [9] as

$$\mu_{t+1} = \mu_t + (\alpha_1(1 - M_t) + \alpha_2 M_t \times D_t) \tag{3}$$

where t is time, $\mu_t$is an updating parameter, andα_1 and α_2 are weights of updating parameters. The variable D_t is the difference between the current value and a parameter model, and the value of M_t is 0 for background pixels and 1 for foreground pixels.

As improvement of the RGA model of Tang, Z et al. [24] combines the RGA model with a frame differing method. Tang and co-authors use frame differing as a post-processing stage to reduce false positive rates and to significantly decrease the detection error by eliminating small gaps and holes. Jabri, S. et al [8] mixed RGA and edge information for each channel and improved the quality and reliability of the results. The edge model built by the applied sable filter on each colour channel yielded horizontal and vertical edge information of an image.

In [22], a new background modelling formula is derived by importing one more adaptive filter and two different updating rates, as described in Eq. (4).

$$B^K_{x.y} = \begin{cases} (1 - \alpha_1)B^{K-1}_{x.y} + \alpha_1 I^K_{x.y} & \text{If } K > 0, AD^K_{x.y} < TH \\ (1 - \alpha_2)B^{K-1}_{x.y} + \alpha_2 I^K_{x.y} & \text{If } K > 0, AD^K_{x.y} < TH \\ I^0_{x.y} & \text{If } K = 0. \end{cases} \tag{4}$$

where B is a background, K is a frame number, AD is an absolute difference and α_1 are updating rates.

Gaussian Mixture model: In many cases, the background may not contain only static objects. Hence, the non-static motions cause varying pixel intensities. The Gaussian Mixture model is a method to model this motion and variation. This model is a developed version of a single Gaussian that clusters each uniform object into several Gaussian distributions [23]. Eq. (5) mathematically presents this model.

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}.\eta(u; \mu_{i,t}, \sigma_{i,t}) \tag{5}$$

where $\eta(u; \mu_{i,t}, \sigma_{i,t})$ is the i-th Gaussian component, $\sigma_{i,t}$ is a standard deviation and $\omega_{i,t}$ is the weight of each distribution. The variable K is the number of distributions and usually varies from three to five depending on the available storage [16].

As Lee, D.S [10] explains in his work, a common problem for the traditional GMM approach is the balancing between speeds and stability. Hence, he introduces a new formula to improve the convergence rate without changing the stability by driving a new adaptive learning rate, as follows

$$learning \quad rate = q_k.(\frac{1-\alpha}{C_k} + \alpha) \tag{6}$$

In Eq. (6), α is a constant and q_k is a posterior probability of a distribution.

Atsushi Shimada et. al [21] introduced the dynamic control of a Gaussian Mixture model by using a dynamic Gaussian component instead of a single constant Gaussian component to improve the accuracy and to reduce the computational time.

2.1.3 Kernel density-based method

The previously mentioned methods use parametric models to estimate the background in the parameter estimation probability distribution built based on the assumption of the pixel intensity or color distribution of images. However, in contrast to the parametric model, a non- parametric model estimates the density function without any prediction about the distribution; hence, it is called distribution-free, and the probability distribution may change from one image to another.

Kernel density estimation with a Gaussian kernel is one of the popular non-parametric approaches introduced by Elgammal, Harwood, and Davis [6]. In this method, the intensity of pixels is modelled by Eq. (7).

$$P_r(x_t) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{(2\Pi)^{\frac{d}{2}}|\sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(x_t-x_i)^T\sigma^{-1}(x_tx_i)} \tag{7}$$

where P_r is a probability function, N is the buffer size, and σ presents the bandwidth. Proper selection of bandwidth is one of the main issues in kernel density estimation methods. A small bandwidth can cause a varying density estimation and too wide of a bandwidth can cause oversampling. However, [6] Elgammal et al introduce formula (8) to calculate the bandwidth, as follows

$$\sigma = \frac{m}{0.68\sqrt{2}} \tag{8}$$

Latter, Noriega and Bernier [13] proposed new non-parametric estimation by combining local kernel histograms and contour features. This approach shows a more stable result in the matter of illumination changes, as contour-based features help to reduce error rates under varying lighting conditions.

In the latest non-parametric approaches, Shengping et al [26] introduce a non-parametric modelling technique that uses spatial and temporal variations of pixels to model the background. To detect a moving object at time t, this method classified each pixel as foreground or background based on a comparison of the probabilities of its neighbouring pixels using a certain threshold. The neighbouring location of the pixel at the i-th row and the j-th column can be obtained by

$$N_{ij} = \{S_{uv}|(u-i)^2 + (v-j)^2 \leq R^2\} \tag{9}$$

where R is the radius of the neighbouring region.

2.2 Thresholding and data validation

Thresholding is a basic procedure in image processing that is designed to segment an object from its background. In this process, pixels in the scene are considered as foreground if their value is greater than a value called the threshold (T), as described in Eq. (10), below

$$F = \begin{cases} F(i,j) > T & \text{object} \\ F(i,j) \leq T & \text{background} \end{cases} \tag{10}$$

Thresholding algorithms can mainly be classified into two broad categories, namely, global thresholding and local thresholding. In global thresholding, a single threshold value is selected for all of the pixels in the entire scene, to segment the objects. In this model, the threshold value can be set empirically [6, 23] to a specific value. In the second case, that of local thresholding, different threshold values are used for different regions in the frame. This model is also called adaptive thresholding and achieves more accurate results in places where the pixel intensity of the background is not uniformly distributed.

In research conducted by Rosin [18], four different thresholding methods are described, based on a distribution model that includes the normal distribution, the intensity distribution, a Poisson distribution and spatial distributions. A worthwhile survey directed by Sankur.B and Sezgin.M [20] contains a review and evaluated most of the common thresholding approaches. This paper classified thresholding techniques into six groups, as follows:

1. Histogram shape-based methods that analyse peaks, valleys and curvature of the histogram and that set the threshold according to these morphological characteristics.

2. Clustering-based methods, where the gray-level samples are clustered into two parts, as background and foreground objects.

3. Entropy-based methods resulting in algorithms that use the entropy of the foreground and the background regions.

4. Object attribute-based methods that search for a measure of similarity between the gray-level and the binary images.

5. Spatial methods that use higher-order probability distributions and correlations between pixels.

6. Local methods that adapt the threshold value on each pixel to local image characteristics.

2.3 Data validation

Designing background subtraction algorithms with increased complexity affects the computational time and memory consumption of the system. Because of these extra costs, the implementation of more sophisticated algorithms is almost impossible on ordinary systems.

The data validation or cleanup process is a group of techniques designed to reduce the error rate that comes about after subtraction. These misclassifications are caused by three critical difficulties in the background subtraction algorithm itself. First, most of the background subtraction algorithms treat pixels independently and ignore any correlations between neighbouring pixels [17]. Second, the adaptation rate may not match the speed of the moving object and last, non-stationary background pixels in the scene can be mistakenly recognised as foreground (such as the shadow of an object or tree branches) [5].

There are various techniques that can be used for data validation; these methods can be divided into five groups, namely blob processing, noise removal, object level feedback, saliency test and optical flow.

Morphological transformations are the main operation in Blob processing; these operations reduce the false positive rate by filling in individual holes and connecting different separate blobs of the same object. Smoothing is the well-known noise removal operation that is used to reduce the noise and camera variations in a system. Object level feedback is a process that updates the background model and can be classified into two categories as conditional updating and unconditional updating. In conditional updating, pixels update after foreground extraction but in unconditional updating, pixels update after the thresholding and subtraction step. The optical flow test is mainly used for distinguishing shadow and ghost from the foreground, based on trajectory analysis of object displacement. The saliency test is based on the assumption that at least some portion of a foreground object should be poorly explained by the background model [2]. Therefore, the saliency test deals with detecting misclassified pixels from the background, in addition fuzzy logic is another approaches to correct foreground and background pixels misclassification in post processing process.

3 Evaluation Methodology

To make a fair comparison among all of the different algorithms explained in this paper, a software implementation was built using MATLAB 9, and the Otsu threshold method was implemented as the thresholding technique based on a built-in MATLAB function. The Intel (R) core (TM) i7-960 @ 3.2 GHz CPU with 5 GB RAM was chosen as the hardware platform. Each algorithm was applied to four different challenging image sequences from the Wallflower data set [7].

The wallflower database is one of the standard datasets for evaluating BGS algorithms. Figure 2 first row shows four image sequences selected from this data base. The first scenario (S1), called "Camouflage", representing colour similarity in situations of foreground and background. In the second scenario (S2), called "waving tree", small movements in the background scene are major obstacles of foreground segmentation. The third scenario (S3), "Time of Day", represents an illumination change in the environment, and last, the forth sequence (S4), called "Bootstrap", displays a crowded area with a complete dynamic scene. These images are in colour format with a size of 120X160. The ground truths of these data sets are manually using Adobe Photoshop for 20 selected frames for each data set.

Low level pixel-based evaluations were computed for each data set and the measures of False Positive (FP), False Negative (FN) and Percentage of Correct Classification (PCC), which were computed for each dataset [19].

The False Positives (FP), or false alarm rate, can be obtained by calculating the number of non-changed pixels that are incorrectly identified as changed pixels. In contrast, the false negatives (FN) rate shows the number of changed pixels that are incorrectly identified as non-changed pixels, and, finally, the percentage of correct classification PCC represents the overall rate of correct identification, which can be expressed according to Eq. (11).

$$PCC = \frac{CD}{CD + FP + FN} \tag{11}$$

In Eq. (11), CD is correct detection (i.e., correct identification), which gives the total number of pixels that are correctly categorised and that can be calculated, as follows

$$CD = Total \quad Pixel - (FN + FP) \tag{12}$$

4 Experimental Results and Discussion

Table 1: Memory usage and speed efficiency for selected algorithms

Algorithm	Speed (frame per second)	Memory Usage (KB)
Median Filtering	0.18 fps	10754 KB
Approximation Median	0.17 fps	1670 KB
RGA	0.05 fps	1094 KB
GMM	0.60 fps	2304KB
KDE	2.00 fps	23962KB

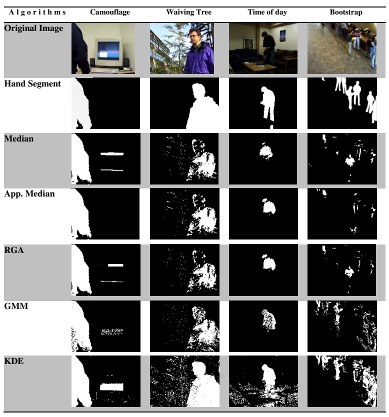


Fig. 2: Comparisons of qualitative results of the selected algorithms. Each column represents a selected frame from different challenging scenarios and application of the BGS algorithms. The first and second rows illustrate the test frames and the ground truth image, and the remaining five rows show the selected algorithms discussed in this paper.

Table 1 contains information about the memory consumption and computational time for the five chosen algorithms. Theoretically, it is expected that will be a huge difference between the three recursive models (RGA, Approximate Median and GMM) and the two non recursive models (KDE, Median Filtering) in the matter of memory usage. This theoretical predication is substantiated because the RGA gives the best performance, with 1094KB of memory consumption, and the KDE gives the worst result, with 23962KB of memory.

Table 2: Quantitative results for selected algorithms in different scenarios

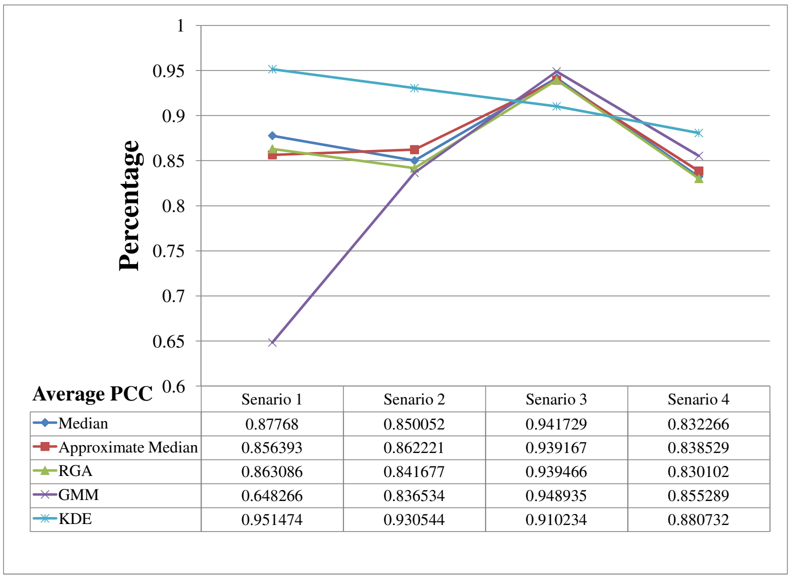
Algorithm	Scenarios	FP	FN	CD
Median	S1	0.005	0.117	0.878
Median	S2	0.014	0.136	0.850
Median	S3	0.000	0.058	0.942
Median	S4	0.018	0.150	0.832
App. Median	S1	0.002	0.142	0.856
App. Median	S2	0.010	0.127	0.863
App. Median	S3	0.000	0.061	0.939
App. Median	S4	0.011	0.150	0.839
RGA	S1	0.004	0.133	0.863
RGA	S2	0.008	0.150	0.842
RGA	S3	0.000	0.061	0.939
RGA	S4	0.019	0.151	0.830
GMM	S1	0.007	0.344	0.649
GMM	S2	0.006	0.158	0.836
GMM	S3	0.000	0.051	0.949
GMM	S4	0.008	0.137	0.855
KDE	S1	0.034	0.014	0.952
KDE	S2	0.058	0.012	0.930
KDE	S3	0.081	0.009	0.910
KDE	S4	0.075	0.044	0.881

With regard to computational speed, once more the RGA is the foremost model, with a speed of 0.05 frames per second, followed by approximate median and median filtering with almost the same computational speed. Again, here the KDE represents the worst case scenario, with a speed of 2 frames per second, which is 4 times slower than GMM.

Here, it is necessary to remark that the Matlab implementations of the mentionned algorithms are not optimised; therefore, the running time is higher than the real time expectation. However, to obtain a unique view of the performances regardless of the hardware and software characteristics, all values can be normalised based on the RGA. Therefore, if we represent the speed of RGA by (R), then Median filtering, Approximate Median, GMM and KDE have speeds of 3.6R, 3.6R, 12R and 40R, respectively.

From our visual observation of Figure 2 and the results obtained from Table 2 and Figure 3 and 4, the following remark can be made:

• The data in Table 2 shows the average rate for the false positive and false negative rates. Moreover, the false positive rate and the false negative graphs are illustrated in Figure 4.1 and Figure 4.2 over 20 selected frames. As can be seen from these figures, the FN error rate of KDE



Average PCC	Senario 1	Senario 2	Senario 3	Senario 4
Median	0.87768	0.850052	0.941729	0.832266
Approximate Median	0.856393	0.862221	0.939167	0.838529
RGA	0.863086	0.841677	0.939466	0.830102
GMM	0.648266	0.836534	0.948935	0.855289
KDE	0.951474	0.930544	0.910234	0.880732

Fig. 3: Percentage of correct classifications in four different scenarios.

is significantly lower compared to other selected models. However, in the FP case, the result of KDE is totally in contrast to the FN rate and gives the weakest performance, this FN error rate in KDE model shows an enormous distinction compared with the other methods. This difference is more significant in the time of day scenario (Figure 4.2 c), where there are gradual changes in lighting conditions.

• The result from Table 2 also proves that the GMM is the second best model after KDE. As shown in Figure 4.3), the GMM gives acceptable performance in 3 out of 4 scenarios. In the third scenario (S3), the GMM obtains the best classification results, with 96% of the PCC value, which implies that gradual illumination changes are better handled by the GMM as compared to KDE and the other algorithms. However, in the first scenario, this model gives the worst performance. The reason for this poor performance can be determined from the attributes of the Gaussian mixture model, which uses intensity values to build a probability density model for the background. Therefore, GMM cannot handle the situation where foreground and background pixels have similar intensities well.

• From the average error rate of median filtering, the approximate median and the RGA in Table 2, it can be seen that these three models give almost the same result in all four different scenarios. This similarity became more obvious with tracking the pattern of these models in Figure 4, where the result is close to identical over a sequence of frames.

5 Conclusions

In this paper, we have evaluated and compared five well-known background modelling algorithms in four challenging situations. The overall evaluation proves the accuracy and effectiveness of

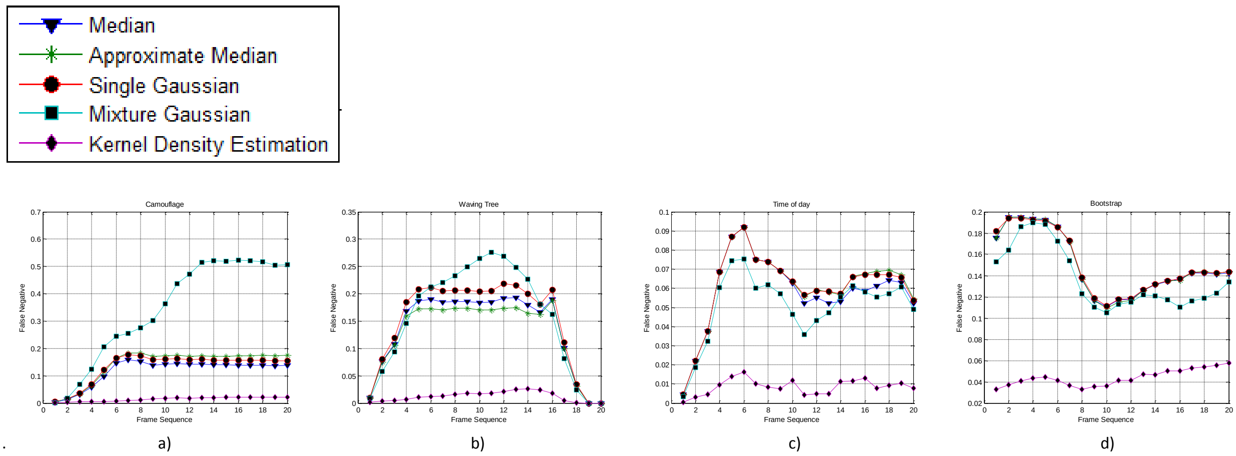


Fig. 4: False negative rate vs. frame sequence for different scenarios: (a) Camouflage (b) Waving tree (c) Time of day (d) Bootstrap

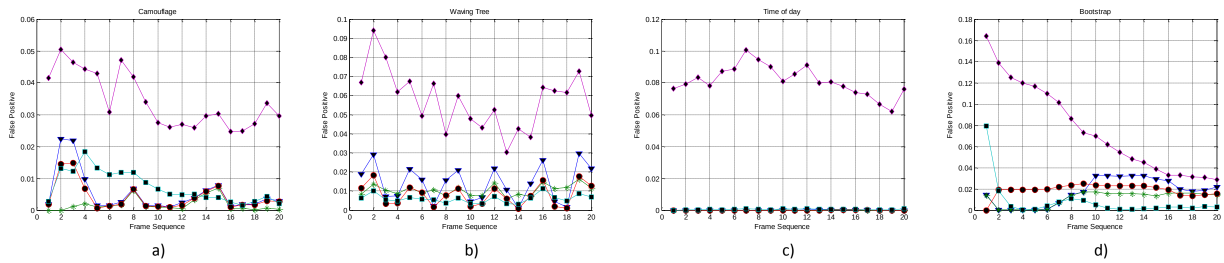


Fig. 5: False positive rate vs. frame sequence for different scenarios: (a) Camouflage (b) Waving tree (c) Time of day (d) Bootstrap

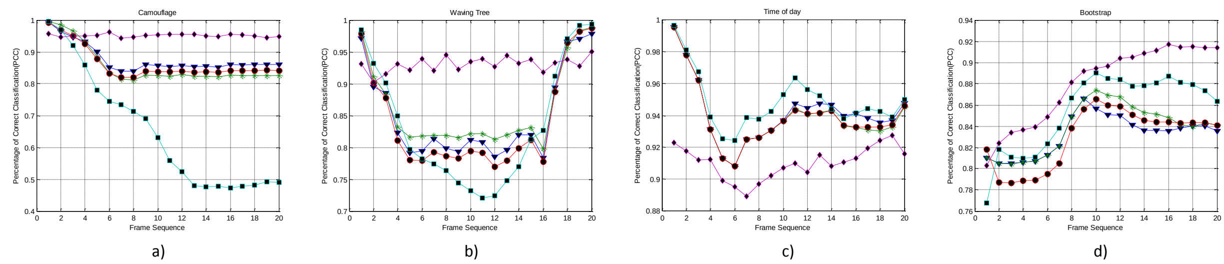


Fig. 6: Percent correct classification vs. frame sequence for different scenarios: (a) Camouflage (b) Waving tree (c) Time of day (d) Bootstrap

KDE, because in 3 out of 4 scenarios, a PCC value of more than 80% was obtained. However, in memory usage and speed efficiency, the KDE is extremely high compared to other methods, which indicates that the KDE is unsuitable for real-time applications. In addition, the RGA and approximate median produce acceptable accuracy with extremely simple implementations, although the accuracy is not as good as KDE.

Finally, as this experimental evaluation shows, in terms of accuracy, no perfect system exists because a perfect system has to solve many problems, such as bootstrapping, illumination changes, and small movements in background and camouflage. However, based on our findings, the Gaussian-based approaches (RGA, GMM) give well-balanced results in speed, accuracy and memory usage for real-time processing applications.

Acknowledgement

This research was supported in part by an eScience Fund grant 01-01-02-SF0563 by MOSTI and OUP-UKM-ICT-36-184/2010 grant from the Centre for Research and Innovation Management (CRIM) of Universiti Kebangsaan Malaysia (UKM).

References

[1] Cheung, S. C. and Kamath, C. "Robust Techniques for Background Subtraction in Urban Traffic Video", Visual Communications and Image Processing San Jose, January 2004, pp. 881 – 892.

[2] Cucchiara, Grana, Piccardi, Prati, "Detecting moving objects, ghosts and shadows in video streams", IEEE Transactions on Pattern Analysis and Machine Intelligence, (vol. 25, no. 10), October 2003, pp. 1337 – 1342.

[3] Cucchiara, Grana, Piccardi, Prati, "Detecting Objects, Shadows and Ghosts in Video Streams by Exploiting Color and Motion Information", in Proc. of ICIAP 2001, 11th International Conference on Image Analysis and Processing, Palermo, Italy, September 26 – 28, 2001, pp. 360 – 365.

[4] Cucchiara, Grana, Piccardi, Prati, "Statistic and Knowledge-based Moving Object Detection in Traffic Scenes", in Proc. of ITSC-2000 — The 3rd Annual IEEE Conference on Intelligent Transportation Systems, Dearborn, MI, USA, Oct. 1 – 5, 2000, pp. 27 – 32.

[5] Elhabian, S., El-Sayed, K., Ahmed, S., Moving Object Detection in Spatial Domain using Background Removal Techniques: State-of-Art Recent Patents on Computer Science Art, Vol. 1 (2008), pp. 32 – 34.

[6] Elgammal, A. M., Harwood, D., and Davis, L. S., Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE (Volume: 90, Issue: 7), Jul 2002, pp. 1151 – 1163.

[7] http://research.microsoft.com/jckrumm/WallFlower/TestImages.htm.

[8] Jabri, S., Duric, Z., Wechsler, H., Rosenfeld, A., "Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information Pattern Recognition", 2000. Proceedings. 15th International Conference, Barcelona, Spain, 3 – 7 Sep 2000, pp. 627 – 630 vol. 4.

[9] Koller, D., Weber, J., Huang, T., Ma, J., Ogasawara, G., Rao, B., Russell, S. "Towards Robust Automatic Traffic Scene Analysis in Real-time". In Proc. of the 12th Int'l Conference on Pattern Recognition (ICPR-94), pp. 126 – 131, Jerusalem, Israel, October 9 – 13, 1994. pp. 126 – 131.

[10] Lee, D. S., Effective Gaussian Mixture Learning for Video Background Subtraction, Pattern Analysis and Machine Intelligence, IEEE Transactions (Volume: 27, Issue: 5), May 2005, pp. 827 – 832.

[11] Lo and Velastin, Automatic Congestion Detection System for Underground Platforms Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on Hong Kong May 2 – 4 2001 on page (s): 158 – 161.

[12] McFarlane N, Schofield C. "Segmentation and Tracking of Piglets in Images", Machine Vision and Applications, (Vol. 8, No. 3). (1 May 1995), pp. 187 – 193.

[13] Noriega, P., and Bernier, O., Real Time Illumination Invariant Background Subtraction Using Local Kernel Histograms. British Machine Vision Association BMVC, Volume: 3, Publisher: Citeseer, (2006), Pages: 1 – 10.

[14] Parks, D. H., Fels, S. S.: Evaluation of Background Subtraction Algorithms with Post-Processing. AVSS '08 Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance IEEE Computer Society Washington, DC, USA, 2008, pp. 192199.

[15] Piccardi, M. "Background Subtraction Techniques: a Review". Systems, Man and Cybernetics, 2004 IEEE International Conference on, 07 March 2005, 3099 – 3104 vol. 4.

[16] Power, p. and Schoonees, J., "Understanding background mixture models for foreground segmentation". Proceedings Image and Vision Computing New Zealand 2002 University of Auckland, Auckland, New Zealand 2628th November 2002, p. 6.

[17] Radke, R. J. Andra, S. Al-Kofahi, O. Roysam, B., Image change detection algorithms: a systematic survey. Image Processing, IEEE Transactions, (Volume: 14, Issue: 3), March 2005, p. 294.

[18] Rosin, P. "Thresholding for change detection". Computer Vision, 1998. Sixth International Conference, 4 – 7 Jan 1998, p. 274.

[19] Rosin, P. and Ioannidis, E., Evaluation of Global Image Thresholding for Change Detection, Pattern Recognition Letters (Volume 24 Issue 14) Elsevier Science, October 2003.

[20] Sankur, B., Sezgin, M., "A survey over Image Thresholding Techniques and Quantitative Performance Evaluation". Journal of Electronic Imaging, (Vol. 13), No. 1). (2004), pp. 146 – 168.

[21] Shimada, A., Arita, D., Taniguchi, R. I., "Dynamic Control of Adaptive Mixture-of-Gaussians Background Model", AVSS '06 Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, IEEE Computer Society Washington, DC, USA, 2006, p. 5.

[22] Shu-Te Su, Yung-Yaw Chen, "Moving Object Segmentation Using Improved Running Gaussian Average Background Model", Computing: Techniques and Applications.DICTA '08. Digital Image, Canberra, ACT, 1 – 3 Dec. 2008, pp. 24 – 31.

[23] Stauffer, Grimson, "Adaptive Background Mixture Models for Real-Time Tracking", Computer Vision and Pattern Recognition, IEEE Computer Society Conference, Fort Collins, CO, USA, 25 Jun 1999, pp. 2246, Volume 2.

[24] Tang, Z., Miao, Z., and Wan, "Background Subtraction Using Running Gaussian Average and Frame Difference", Entertainment Computing — ICEC 2007, 6th International Conference, Shanghai, China, September 15 – 17, 2007. pp. 411 – 414.

[25] Wren, C., Azarhayejani, A., Darrell, T., Pentland, A. P. "Pfinder: real-time tracking of the human body". Automatic Face and Gesture Recognition, Proceedings of the Second International Conference on. Killington, VT , USA. 16 Oct 1996, pp. 51 – 56.

[26] Zhang, S., Yao, H., Liu, S., "Spatial-Temporal Nonparametric Background Subtraction in Dynamic Scenes", Multimedia and Expo ICME 2009. IEEE International Conference, New York, NY, July 3 2009, pp. 518 – 521.