

A Robust Median-based Background Updating Algorithm

ObedAppiah

Department of Computer Science and Informatics University of Energy and Natural Resources
Email: obed.appiah@uenr.edu.gh

James Ben Hayfron-Acquah

Department of Computer Science Kwame Nkrumah University of Science and Technology
Email: jbhayfron-acquah.cos@knust.edu.gh

Abstract—Image processing techniques for object tracking, identification and classification have become common today as a result of improved quality of cameras as well as prices of cameras becoming cheaper and cheaper day by day. The use of cameras also make it possible for human analysis of video streams or images where it is difficult for robots or algorithms or machines to effectively deal with the images. However, the use of cameras for basic tracking and analysing do not come without challenges such as issues with sudden changes in illumination, shadows, occlusion, noise, and high computational time and space complexities of algorithms. A typical image processing task may involve several subtasks such as capturing, and pre-processing which demand high computational resources to complete. One of the main pre-processing tasks used in image processing is image segmentation which enables images to be divided into sections of interest in order to perform analysis on them. Background Subtraction is commonly used to segment images into Background and Foreground for further processing. Algorithms producing highly accurate results during this segmentation task normally demand high computation time or memory space, while algorithms that use smaller memory space and shorter time to complete this segmentation task may also suffer from limitations that may lead to undesired results at some point in time. Poor outputs from algorithms will eventually lead to system failure which must be avoided as much as possible. This paper proposes a median based background updating algorithm which determines the median of a buffer containing values that are highly correlated. The algorithm achieves this by deleting an extreme value from the buffer whenever data is to be added to it. Experiments show that the method produces good results with less computational time which will make it possible to implement on devices that do not have much computation resources.

Index Terms—Image processing, background updating, background subtraction model, approximated median filter

I. INTRODUCTION

Image processing technique for object tracking, identification and classification has become common today as a result of improved quality of cameras as well as prices becoming cheaper and cheaper day by day. The use of cameras also make it possible for human analysis of video streams or images where it is difficult for robots or algorithms or machines to effectively deal with the images. Research works in image processing primarily are looking for new and effective ways of fully using cameras to implement task that are done by humans in order to automate processes. However, the use of cameras for basic tracking and analysing do not come without challenges such as issues with illumination, shadows, occlusion, and noise. Various algorithms have been proposed to deal with these challenges in order to have robots able to compete with their human counterparts with high levels of competences. Even though these challenges are being addressed by various researchers, one main challenge still persists within the domain of image processing and that is how to improve the performance of algorithms or design new ones so that they can be very efficient with respect to time and space complexities [1, 2]. The reason for continuous work is to have algorithms that run very fast and take shorter computational time to complete tasks, notwithstanding the fact that results produced by these algorithms must be accurate for the tasks they are designed for [3].

One of the main tasks performed for almost all image processing task is image segmentation, which enables an algorithm to divide an image into two or more components. It is usually the first task of any image analysis process module and thus, subsequent tasks rely strongly on the quality of segmentation [4, 5, 6]. Segmentation processes simply seeks to evaluate pixel values to see whether they fall within a category or not, usually done by comparing pixel values with some threshold value(s). Marykutty and Sankar in [7] used the object segmentation method for content based video management and compression of video frames for video conferencing. The face region, which is the object of interest in the video frames, is identified first using a skin

colour based algorithm. This algorithm was applied so that an image will generate face and non-face pixels in order to apply different compression technique to them. One of the most common segmentation processes is the Background Subtraction, which enables the system to derive two images from a given set of images or video stream normally referred to as Background and Foreground. In all cases the background is normally considered as the element in the scene that is not of interest to the system and the foreground is the set of pixels the system will be interested in for further processing. Various algorithms have been proposed to deal effectively with this task of segmenting the background (static section in a video stream) and foreground (not-static section in a video stream), but not without one constraint or the other. Algorithms producing highly accurate results normally demand high computation time or memory space, while algorithms requiring smaller memory space and shorter time to complete may also suffer from limitations that may lead to undesired results at some point in time [8]. Poor outputs from algorithms will eventually lead to system failure. Another interesting observation is that, it is very difficult to have an ideal algorithm that can fit all situations or scenarios, hence the need to have one that best fit most situations in order for effective application of image processing techniques for tracking, identification, classification, and recognition [9].

Currently, background subtraction is a common method to detect objects [10, 11] or to separate the foreground from the background. A background subtraction is easy and fast for tracking objects, however extracting and updating the background in real time is the critical task of this method or algorithm or model, which demands a lot of computational time. As a result, researchers have proposed many methods to perform this critical task since an effective one will always lead to better object detection in complex environments or scene. Complex scenes occur as a result of changes in light intensity, snow, rain and other environmental factors [12]. The strength of the background model depends on how to effectively update the background when such environmental factors occur. Many background updating methods have been proposed in the past decades including Running Gaussian Average, Temporal Median Filter, Mixture of Gaussian, Kernel Density Estimation (KDE), and Kalman Filter.

II. RELATED WORKS

The main challenges in the domain of image processing have always been the time and space complexities of algorithms. Interestingly there is not a single algorithm that can fit all situations as far as background updating is concern. One algorithm may have a very good time complexity, but may be poor in terms of handling background updating values effectively. Another algorithm that may generate a very good or excellent output or background updating values may suffer from high time complexity to get the job done

successfully. For example, the Mixture of Gaussian (MoG) method is among the best performing background subtraction algorithms but suffers from high computational time and its parameters require careful tuning. Secondly, it is very sensitive to sudden changes in global illumination. If a scene remains stationary for a long period of time, the variances of the background components may become very small. A sudden change in global illumination can then turn the entire frame into foreground [11]. Stauffer et al. [13, 14] proposed Gaussian Mixture Model (GMM) to model the value of each pixel of the video frames as a mixture of Gaussians. This method can solve the problems of light changes and waving trees in the background, but the parameters of the Gaussian distributions need to be estimated and optimized in advance, which will lead to high computational complexity. This challenge makes it difficult to actually implement this method for real-time application especially on microcontrollers and low-end devices. Again the online updating of the model is so slow [15] that this method cannot precisely detect large objects which are moving slowly [16].

Temporal Median Filtering method of updating background values is one of the simplest methods used to model background. Lo and Velastin [17] proposed to use the median value of the last n frames as the background model. The method works well but performance is usually based on the size of the buffer and the nature of video stream to be analysed. It has been implemented successfully in areas of tracking objects in a simple video stream. In addition, Cucchiara [18] proposed to compute the median on a special set of values containing last n , sub-sampled frames and w times the last computed median value. This combination increases the stability of the background model. Piccardi [10] concluded that the main disadvantage of the median-based approach is that its computation requires a buffer with the recent pixel values. Another disadvantage of the median-based is its method used to select the median for the updating of the background. Parameter and Non-parametric methods of selection of median values have been proposed, and most time the non-parametric methods have shorter computational time which generally improves the performance of the algorithm [19, 20]. Again, the median filter does not accommodate for a rigorous statistical description and does not provide a deviation measure for adapting the subtraction threshold [10].

Based on approximated median filtering (MF), McFarlane [13] proposed a background model which stores only one video frame that is the background image for object detection. The strength of the algorithm is that it has a very low computational time as compared to most of the background updating algorithms and again does not require a buffer to store the last n frames. This method is useful for the small objects moving rapidly. However, when the objects are large and moving slowly there is ghost of the objects in the background and obviously it will affect the detection results. Especially when moving objects slow down and stop finally, the objects will merge into the background as background

updating and cannot be detected. This method can be improved by using bootstrapping to determine the first set of pixels to be considered as Background pixel. Our experiments showed that the initial value considered as background pixel will affect subsequent background pixel estimation. Approximated approach used by Remagnino et al [22] also generated good result but not as compared to MoG.

Sen-Ching et al [11] concluded in their experiment that MoG and MF produced better results than that of approximated median filters as well as Kalman Filter, but MoG has high time complexity while MF has high space complexity. That is, the median-based background updating models uses only the last n frames for estimation and selection of median values using the sorting method also requires high computational time. This paper proposed another median based background updating model which determines the median of a buffer of size n which contains values that are highly correlated as compared to the traditional means of estimating the median values.

III. THE PROPOSED ALGORITHM

The proposed algorithm uses a buffer of size n to store n images from which the median values for all corresponding locations (x,y) for all images are estimated and stored as background. Unlike the traditional median filtering which uses a sliding window approach to update the background, this algorithm does not use this method, but still maintain a sorted list in order to generate the value to be used for updating the background. That is, the proposed algorithm performs pixel value insertion into the buffer and deletes either the least value or maximum value in the buffer. This makes it possible for impulse noise to be quickly removed before it influences the selection of the median for updating the background as seen in some cases of the median filter updating method and some of the approximated median-base background updating models.

A. The key features of the proposed algorithm are

- i. It is a median-based algorithm which makes it simple and acceptable since median based algorithms have been widely used successfully [23].
- ii. Sorting is only done during the bootstrapping period, after which there will be no need to re-sort the buffer again.
- iii. Time complexity for the algorithm is much faster than the Temporal Median Filter algorithm.
- iv. The set of values from which the median is selected usually exhibits a better correlation among elements than that of the temporary median filter algorithm, which means updating pixel values generated by the proposed are relatively better.
- v. During the insertion of a value into the buffer, an extreme value is deleted. The value to be deleted depends on the magnitude of the incoming value.

The buffer is logically divided into two(2) sections, the upper and the lower. The upper section contains values that are all larger or equal to that of the lower section. Insertion operations must always ensure that the list is sorted so that the median can easily be determined. If a value is to be inserted into the lower section, then the maximum value in the buffer will be deleted and vice versa. This makes it possible for the proposed algorithm to handle noisy pixels (impulse noise) well and avoids using them when updating the background.

- vi. Binary search algorithm is used to find where a value is to be inserted, and the time taken to perform this task is $O(\log n)$
- vii. The shifting of values in order for the insertion operation to be performed takes $O(n)$ for worst case scenario.
- viii. The selection of the median value for size n buffer will take $O(n) = \{O(\log n) + O(n)\}$ for worst case scenario, which performs much better than the traditional median filter approach.

B. The Algorithm

- i. Create a buffer of (x, y, n) size
- ii. Read the first n images into the buffer
- iii. Sort the content of the buffer and Select Buffer $(x, y, (n-1)/2)$ as value to be used for updating the background.

$$B(x, y) = \text{Buffer}(x, y, (n-1)/2)$$

- iv. Read next Image I_t
- v. Determine where $I_t(x,y)$ can be inserted into the buffer using Binary Search method.
- vi. Determine whether the location is in the upper section of the list or lower section of the buffer.
- vii. Shift data to the right if the value is less than the current median value; otherwise shift data to the left. That is, the smaller values will be lost when the incoming values are bigger than the current median value, maximum values are deleted otherwise.
- viii. Select the median value and update the background with it $B(x, y) = \text{Buffer}(x, y, (n-1)/2)$

C. Illustration of How the Proposed Algorithm Works

Figure 1 illustrates how the algorithm works using a given sample values and a buffer of size 11 cells. The buffer is initially filled and sorted. To add value 14 to the buffer, the location to insert the value is first searched and identified. Since 14 will have to be added to the lower part of the buffer, the maximum value is deleted, all data greater than 14 are shifted to the right and the value (14) inserted. The median value is selected as the background updating pixel value. The second illustration inserts value 27 to the list and selects the median for updating the background.

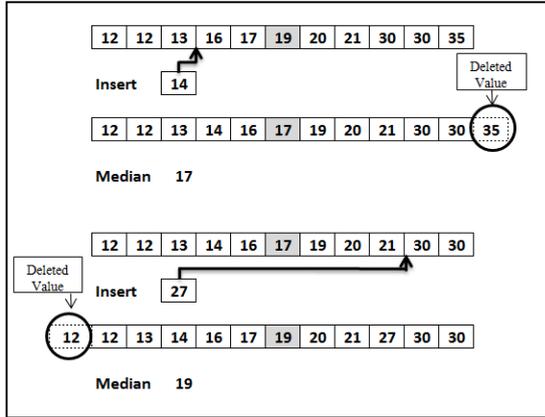


Fig.1. Inserting value 14 and 27 into buffer

IV. TESTING OF THE ALGORITHM

Three (3) main tests were used to evaluate the effectiveness of the proposed algorithm. Generally the strength of the proposed algorithm is that it is faster when compared to the traditional median method for background updating algorithm and also in terms of updating the background pixels, the selected values are always closer to the ground-truth. Again as a result of the insertion procedure described above, the method is able to delete extreme values, which gives it the property of inherently dealing with impulse noise found in captured images. These tests or analysis performed was to measure how these claims could be ascertained.

Randomly generated values test, performance of algorithm on selected images and Big-O analysis to determine time complexity of the proposed algorithm were used to test the algorithm.

A. Randomly Generated Values Test.

Ten thousand (10,000) randomly generated values between 0 and 255 were used for this testing. The value 128 was selected as the ground-truth background pixel or value, and the algorithms were applied to the given set of data to determine which algorithm generated background updating value equal to the selected ground-truth value. Nine different sets of data were generated randomly with each set containing 10,000 values. The percentage of the selected ground-truth values were varied for each set of data before the algorithms were made to generate their updating values. The number of times an algorithm

generated 128 as background updating value was counted for analysis. The General method was as follows:

- i. Determine the estimated background pixel value at any given time. (b_t)
- ii. Count how many estimated (b_t) are equal to 128 (the selected Ground-truth).
- iii. Algorithm that produced higher frequency for the counting technique was selected as a good candidate for generating background updating values.

B. The Image Test

The second experiment was performed by using a sampled recorded video stream to check for some of the strength and weakness of the proposed algorithm. This technique for testing the performance of image processing algorithm is quiet common since researchers are able to present visuals of the outcome of their algorithms and able to compare with others. The proposed algorithm was implemented using MATLAB and tested with sample video. Three other algorithms (temporary median filtering, running average and Mcfarlane’s median approximation) were also tested with the same sample video. During the testing, the rate of extraction of the frames from the video stream was also varied to simulate the common scenarios of fast and slow moving objects in a scene. Two (2) main categories of sample rate performed to mimic slow (6 frames per second), and fast (2 frames per second) moving objects in the scene respectively.

C. Computation Complexities

The final test evaluated the time and space complexities of the proposed algorithm and that of the three other median algorithms. The Big O Notation was used to estimate the computational time of the algorithms.

V. RESULTS

A. Results for Randomly Generated Values Test

Table 5.1 illustrates the various percentages of ground-truth values in the 10,000 randomly sampled data and the number of times an algorithm’s estimate matched the ground-truth.

Table 1. Number of estimated background updating values same as 128 for the various algorithms.

Percentage Distribution of Ground-truth Value (128)	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Running Average	98	116	114	188	353	498	896	1663	2829	9980
McFarlane	1036	2150	3136	4106	4995	6000	6996	7983	8995	9980
Median Filter	3606	6496	8537	9494	9895	9949	9980	9980	9980	9980
Proposed App. Median Filter	6604	9658	9962	9980	9980	9980	9980	9980	9980	9980

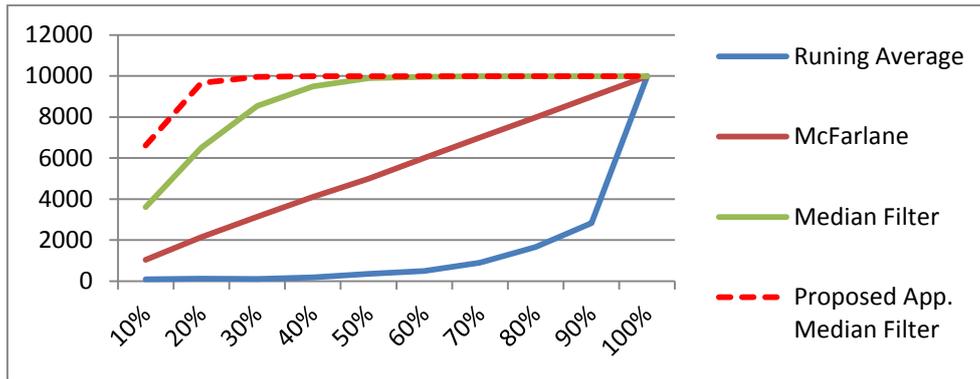


Fig.2. A graphical representation of Table 1

Table 1 indicated that the proposed algorithm for most of the time predicted or generated background updating values that were the same as the ground-truth for all the distributions. This suggested that the method can do well

when used for selecting background updating values; however it must be stated that the distribution of the randomly generated values can also affect the performance of this method.

B. Results for Sample Image Tests

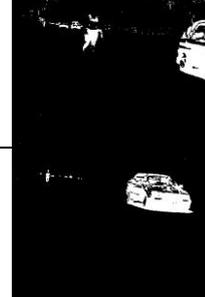
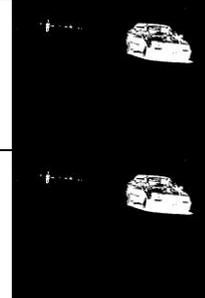
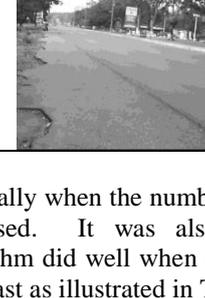
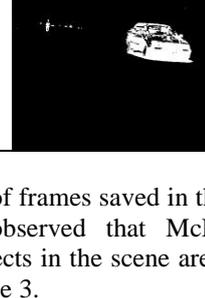
Selected frames from a video stream



Fig.3.

Table 2. Performance of various algorithms on relatively slow moving objects in scene, Two (2) Frames per second

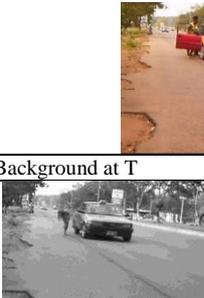
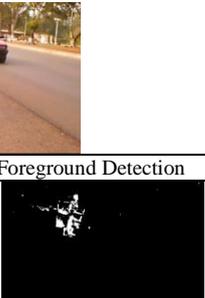
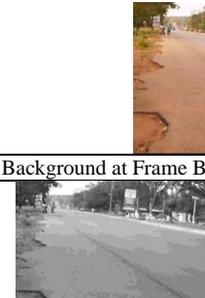
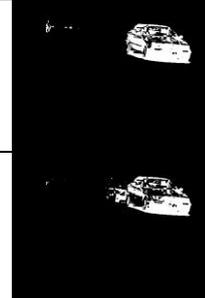
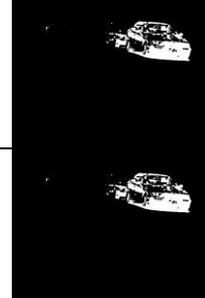
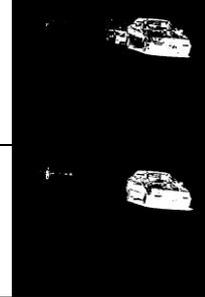
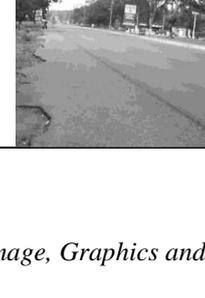
	Frame A		Frame B	
	Background at T	Foreground Detection	Background at Frame B	Foreground Detection
Median				
Running Average ($\alpha = 0.05$)				
Running Average ($\alpha = 0.3$)				

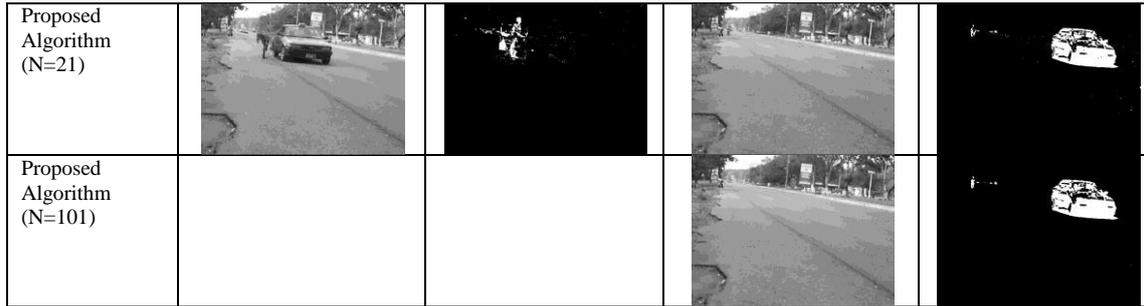
Running Average ($\alpha = 0.7$)				
McFarlane				
Proposed Algorithm (N=21)				
Proposed Algorithm (N=101)				

The sample images in Table 2 indicated that the proposed algorithm performed relatively better in trying to identify an object in the background. The issue of ghost was much minimized with the proposed algorithm

especially when the number of frames saved in the buffer increased. It was also observed that McFarlane’s algorithm did well when objects in the scene are moving very fast as illustrated in Table 3.

Table 3. Performance of various algorithms on relatively fast moving objects in scene Six (6) Frames per second

	Frame A		Frame B	
	Background at T	Foreground Detection	Background at Frame B	Foreground detection
Median				
Running Average ($\alpha = 0.05$)				
Running Average ($\alpha = 0.3$)				
Running Average ($\alpha = 0.7$)				
Macfarlane				



From the above Table 3, Macfarlane’s approximated median works very well just like the proposed. This was as a result of the rate of movement of objects in the scene.

C. Time Complexity Analysis of the Proposed Algorithm.

Table 4 illustrates Time and Space Complexities of the proposed algorithm together with other median-based background modelling algorithms.

Table 4. Time and Space complexities of the proposed algorithm and other algorithms.

Algorithm	Time Complexity	Space Complexity
Median	$O(n^2) / O(n)$	$O(n)$
Running Average	$O(1)$	$O(1)$
Macfarlane	$O(1)$	$O(1)$
Proposed Algorithm	$O(n)$	$O(n)$

VI. CONCLUSION

The idea of having a single algorithm that best fit for background modelling has always been a challenge. A algorithm may do well in one scenario and may performbadly in another scenario. Very good algorithms used in modelling background do have high computational or space complexities. The practicehas always been to anticipate the type of scene ideal for a typical algorithm, and develop system based on them in order to get a job done. The research proposed a new median based algorithm which can generally handle various scenarios and can be used to generate good results at all times even when the nature of scenes to be used by the application is not known in advance.

REFERENCES

[1] Jian Z. H. C., Qingwei Liang "A Novel Method for Traffic Object Detection Based on Improved Approximated Median Filter" Journal of Information & Computational Science 9: 8 (2012) 2253–2261

[2] AlawiM. A., Othman O. Khalifa and M.D. Rafiquel Islam; Performance Comparison of Background Estimation Algorithms for Detecting Moving Vehicle, World Applied Sciences Journal 21 (Mathematical Applications in Engineering): 109-114, 2013 ISSN 1818-4952 © IDOSI Publications, 2013 DOI: 10.5829/idosi.wasj.2013.21.mae.99934

[3] Mao-Hsiung H., Jeng-Shyang Pan, Chaur-Heh Hsieh "A Fast Algorithm of Temporal Median Filter for Background Subtraction" Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212 Volume 5, Number 1, January 2014

[4] YonghongQ., Zhou Shshenqi, "Review on Uniform color space and Color Difference Formula", Print World, 2003.9:16-19.

[5] Nilima K., "ColorThresholding Method for Image Segmentation of Natural Images" I.J. Image, Graphics and Signal Processing, 2012, 1, 28-34

[6] Ashraf A. Aly, Safaai Bin Deris, NazarZaki "Research Review For Digital Image Segmentation Techniques", International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 5, Oct 2011 DOI : 10.5121/ijcsit.2011.3509 99

[7] Marykutty C., Sankar. P.,"An Object of Interest based Segmentation Approach for Selective Compression of Video Frames", International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol.8, No.2, pp.37-44, 2016.DOI: 10.5815/ijigsp.2016.02.05Published Online February 2012 in MECS (<http://www.mecs-press.org/>) DOI: 10.5815/ijigsp.2012.01.04

[8] AlawiM. A., Othman O. Khalifa and M.D. Rafiquel Islam; Performance Comparison of Background Estimation Algorithms for Detecting Moving Vehicle, World Applied Sciences Journal 21 (Mathematical Applications in Engineering): 109-114, 2013 ISSN 1818-4952 © IDOSI Publications, 2013 DOI: 10.5829/idosi.wasj.2013.21.mae.99934

[9] TangZ., Z. Miao, Y. Wan, "Background Subtraction Using Running Gaussian Average and Frame Difference", IFIP International Federation for Information Processing 2007

[10] PiccardiM., "Background subtraction techniques: A review", in: Proceedings of the International Conference on Systems, Man, and Cybernetics, 2004, 3099-3104

[11] Sen-Ching S. Cheung, Chandrika Kamath, Robust techniques for background subtraction in urbantraffic video, in: Proceedings of Visual Communications and Image Processing 2004, 2004, 881-892

[12] Su L., H. Chen, "Video-background update based on Median Filtering, Opto-ElectronicEngineering", 37(1), 2010, 131-135

[13] McFarlaneN., C. Schoeld, "Segmentation and tracking of piglets in images, Machine Vision andApplications", 8(3), 1995, 187-193

[14] StaufferC., W. E. L. Grimson, "Learning patterns of activity using real-time tracking", IEEE Transactionson Pattern Analysis and Machine Intelligence, 22(8), 2000, 747-757

[15] FengH., S. Gong, C. Liu, "Foreground detection based on improved Gaussianmixture model", Computer Engineering, 37(19), 2011, 179-182

- [16] MaY., W. Zhu, S. An, "Improved moving objects detection method based on Gaussian mixture model", *Computer Applications*, 27(10), 2007, 2544-2548
- [17] LoB.P.L., S.A. Velastin, "Automatic Congestion detection system for underground platforms", *Proc. ISIMP 2001*, pp. 158-161, May 2001
- [18] CucchiaraR., C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams" *IEEE Trans on Pattern Anal. And Machine Intell.*, vol 25, no.10, pp. 1337-1442, 2003
- [19] ElgammalA., D. Harwood, L. Davis, "Non-parametric model for background subtraction", in: *Proceedings of the 6th European Conference on Computer Vision*, 2000, 751-767
- [20] ElgammalA., R. Duraiswami, D. Harwood, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", in: *Proceedings of the IEEE*, 90(7), 2002, 1151-1163
- [21] StaufferC., W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, 246-252
- [22] RemagninoP., A. Baumberg, T. Grove, D. Hogg, T. Tan, A. Worrall, K. Baker, "An integrated traffic and pedestrian model-based vision system," in *Proceedings of the Eighth British Machine Vision Conference*, pp. 380, 389, 1997.
- [23] LuN., J. Wang, Q.H. Wu and L. Yang, "An Improved Motion Detection Method for Real-Time Surveillance", *IAENG International Journal of Computer Science*, 35:1, IJCS_35_1_16

Authors' Profiles



Obed Appiah received the BSc degree in Computer Science from the Kwame Nkrumah University of Science and Technology (KNUST), Kumasi, Ghana in 2005, and his MPhil Computer Science degree from the same institution in 2013. He is currently a Lecturer at the University of Energy and Natural Resources (UENR) - Sunyani, and a PhD Computer Science student at KNUST.



Dr James Ben Hayfron-Acquah is a Senior Lecturer. He received the BSc degree in Computer Science from the Kwame Nkrumah University of Science and Technology (KNUST), Kumasi, Ghana in 1991, his MSc Computer Science and Applications degree from Shanghai University of Science and Technology, Shanghai, China in 1996 and his PhD from the Southampton University, Southampton, England in 2003. He is currently a Senior Lecturer and the Head at the Department of Computer Science, KNUST. He has over 50 publications to his credit. His main research areas include Biometrics, Cloud Computing, Image Processing and Computer Vision.

How to cite this paper: Obed Appiah, James Ben Hayfron-Acquah, "A Robust Median-based Background Updating Algorithm", *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, Vol.9, No.2, pp.1-8, 2017. DOI: 10.5815/ijigsp.2017.02.01