Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Application

Mohamad Hoseyn Sigari[†], Naser Mozayani[†] and Hamid Reza Pourreza[††]

[†]Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran
[††]Computer Engineering Department, Ferdowsi University of Mashad, Mashad, Iran

Summary
Running average method and its modified version are two simple and fast methods for background modeling. In this paper, some weaknesses of running average method and standard background subtraction are mentioned. Then, a fuzzy approach for background modeling and background subtraction is proposed. For fuzzy background modeling, fuzzy running average is suggested.
Background modeling and background subtraction algorithms are very commonly used in vehicle detection systems. To demonstrate the advantages of fuzzy running average and fuzzy background subtraction, these methods and their standard versions are compared in vehicle detection application. Experimental results show that fuzzy approach is relatively more accurate than classical approach.

Key words:
Fuzzy Background Modeling, Fuzzy Background Subtraction, Fuzzy Running Average, Vehicle Detection.

1. Introduction

Background extraction is an important part of moving object detection algorithms that are very useful in surveillance systems. Moving object detection algorithm will be simple when a clean background image is available. Method of background extraction during training sequence and updating it during input frame sequence is called background modeling. The main challenges in moving object detection is extraction a clean background and its updating. There are various methods for background modeling. Some of these methods such as mean filter [1] and median filter [2] need very huge memory capacity and some other such as Eigen-background [3] and Mixture of Gaussian (MOG) [4,5] have more computational complexity.

1.1 Background Subtraction

When the background image obtained, moving objects of the scene can be detected using background subtraction. By applying a threshold on absolute difference of current image frame and background image, moving objects can be detected. Following equation shows background subtraction formula.

$$BGS(i,j)=\begin{cases}1 & if\ \left|I_t(i,j)-BG_{t-1}(i,j)\right|>th_s \\ 0 & otherwise\end{cases} \quad (1)$$

In this equation, I_t is input frame at time *t*, BG_{t-1} is background image at time *t-1* and *BGS* is the result of background subtraction. For a given pixel at location (i,j), if the result of background subtraction was zero, this pixel is a part of scene; else, it is a pixel of moving object.

1.2 Running Average

The commonly, fastest and the most memory compact background modeling is running average method. In this method, background extraction is done by arithmetic averaging on train sequence. After background extraction, background may change during detection of moving objects. Illumination changes are an important reason of background changes. Because of scene illumination change and some other reasons, background image must be updated in each frame. In running average method, background is updated as follow:

$$BG_t=\alpha.BG_{t-1}+(1-\alpha).I_t \quad (2)$$

In this equation α must be in range *(0,1)*, but its rational values have to in range *(0.5,1)*. If α is close to *0.5*, this method will be called short-term running average and if α is close to 1, this method will be called long-term running average.
From signal and system point of view, equation (2) is an Infinite Impulse Response (IIR) filter. Therefore, running average method is an IIR system. But, because of low computational complexity and high memory compactness, running average method is used in real-time systems yet [6,7,8]. In this paper, our goal is not to eliminate this property of running average method; our goal is to improve performance of running average method using fuzzy theory.
A modified running average method for background updating is as follow:

$$BG_t=\begin{cases}\alpha.BG_{t-1}+(1-\alpha).I_t & if\ \left|I_t(i,j)-BG_{t-1}(i,j)\right|<th_u \\ BG_{t-1} & otherwise\end{cases} \quad (3)$$

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

Table 1: An example of using standard background subtraction and modified running average method

time (frame #)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Input gray level	100	129	129	129	100	100	100	134	134	134	100	100	100	139	139
background value	100	102.9	105.5	107.9	107.1	106.4	105.7	108.6	111.1	113.4	112.1	110.9	109.8	112.7	115.3
detection status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
time (frame #)	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Input gray level	139	100	100	100	82	82	82	100	100	100	100	100	80	80	80
background value	117.7	115.9	114.3	112.9	112.9	112.9	112.9	111.6	110.4	109.4	108.5	107.6	104.9	102.4	100.1
detection status	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

th_u is update threshold and must be less than or equal with th_s . When $th_u = th_s$, background image will be updated if the pixel is not detected as moving object. Modified running average method has better performance and better result with respect to standard running average method. Nevertheless, it has some drawbacks because of using hard limiter function in background subtraction and background updating.

1.3 Some Weakness of Standard Background Subtraction and Running Average Method

Some weaknesses of standard background subtraction and modified running average method are being shown in an example. Assume the gray level of real background for a given pixel $I_t(i_0, j_0)$ is $BG_t(i_0, j_0) = 100$ and the gray level of real background has estimated without error in background extraction phase. Now, assume an input sequence $I_t(i_0, j_0) =$ *{100, 129, 129, 129, 100, 100, 100, 134, 134, 134, 100, 100, 100, 139, 139, 139, 100, 100, 100, 82, 82, 82, 100, 100, 100, 100, 100, 80, 80, 80}*, for *t=1* to *30*, $\alpha = 0.9$ and $th_u = th_s = 30$. There are five objects in the input sequence. Gray levels of these objects are *129*, *134*, *139*, *82* and *80* respectively. In the first view, it seems that the first, fourth and fifth objects that has been appeared at *t=2* to *t=4*, *t=20* to *t=22* and *t=28* to *t=30* will not be detected. Also two other objects that have been appeared at *t=8* to *t=10* and *t=14* to *t=16* will be detected by background subtraction.

Table 1 explains this example more. The prejudgment is not true and only forth object will be detected as a moving object. Although gray level of forth object is *82* and its difference from real background gray level is less than th_s . In addition, we expected that fifth object would be detected, but fifth object will not be detected.

The main reason of this unexpected behavior is in gray level of first object. Gray level of first object was only 1 level less than th_s , so it was not detected. In addition, it has damaged background gray level.

In running average method, if background image is damaged, it will difficult to repair background, unless the real background is appeared for a long time without any moving object in scene. So, background updating must be performed accurately.

2. Fuzzy Background Subtraction

In standard background subtraction method, a hard limiter function is used to determine a pixel is a moving object pixel or no. We proposed to use a saturating linear function instead of hard limiter in fuzzy background subtraction.

$$FBGS(i,j) = \begin{cases} 1 & if\ \left|I_t(i,j) - BG_t(i,j)\right| > th_s \\ \dfrac{\left|I_t(i,j) - BG_t(i,j)\right|}{th_s} & otherwise \end{cases} \quad (4)$$

So, the result of fuzzy background subtraction ($FBGS$) will be a real value in range *[0,1]*. In real world, the background subtraction output must be true (foreground) or false (background). To determine a crisp value for output, we propose binarization of fuzzy background subtraction after passing through a low pass filter (LPF). In simple mode, a *3x3* mean filter can be used as a fast low pass filter. Therefore, final output can computed as below:

$$BGS(i,j) = \begin{cases} 1 & if\ \left|LPF(FBGS(i,j))\right| > th_{fs} \\ 0 & otherwise \end{cases} \quad (5)$$

Where, the th_{fs} is a threshold that determines do we perform background subtraction or not. Applying a threshold after low pass filtering offers us more reliability and robustness in moving object detection. This method can detect moving objects that their gray level is similar to background gray level. In addition, it can remove small noise, because of low pass filter existence.

3. Fuzzy Running Average

Fuzzy theory can be used in running average method to update background image. In fuzzy running average method, α is not an overall value. It is defined for each pixel based on current value of fuzzy background subtraction. Rational value for α have to be in range *(0.5,1)*. Therefore, we propose following equation to compute α in each pixel.

$$\alpha(i,j) = 1 - (1 - \alpha_{\min})\exp(-5 * FBGS(i,j)) \quad (6)$$

In this equation $\alpha_{\min}$ is the minimum value for α. This equation is not unique, so it can be changed based on application. For real-time computation, it is better to implement α as a look-up table. Fig. 1 shows diagram of α - $FBGS$ for $\alpha_{\min} = 0.9$.

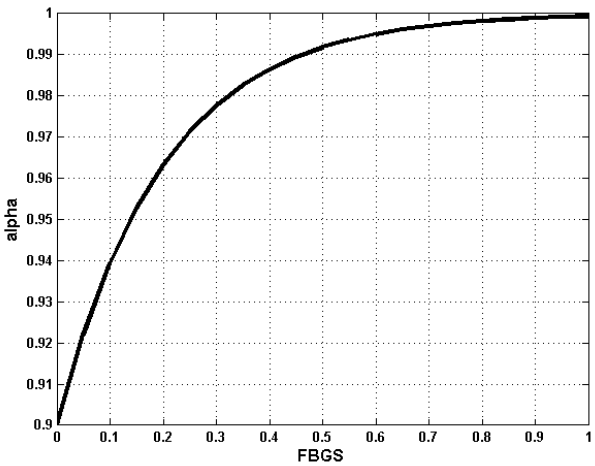

Fig. 1 Diagram of α - $FBGS$ for $\alpha_{\min} = 0.9$.

Background updating in a given pixel using fuzzy background subtraction will be defined as below:

$$BG_t(i,j) = \alpha(i,j).BG_{t-1}(i,j) + (1-\alpha).I_t(i,j) \qquad (7)$$

To show advantages of fuzzy background subtraction and fuzzy running average, last example is being performed again by fuzzy approach. This example was defined only for one pixel not for a real frame sequence. Therefore, performing the LPF after fuzzy subtraction was ignored. In this example we use $th_s = 30$ and $th_{fs} = 1$. The result of fuzzy background subtraction and fuzzy running average is shown in Table 2. Detection status of frame sequence has no error using this fuzzy approach. In other word, objects with gray level higher than *130* or lower than *70* were detected as foreground (moving object). In addition, the background values are very near to real background value (*100*).

4. Vehicle Detection using Fuzzy Background Modeling and Fuzzy Background Subtraction

One of important applications of background modeling is in Vehicle Detection Systems (VDS). VDSs are used in many traffic systems for vehicle counting [9], vehicle classification [10], surveillance [11], traffic parameter extraction [9,12], exerting the traffic rules and vehicle navigation systems [13]. VDSs have to process input frame sequence in real-time usually on a general-purpose processor.

Generally, VDSs [9,10,11,12,14] use background modeling and background subtraction techniques to detect vehicles, because the other techniques have more computational complexity. In vehicle detection, extensive illumination changes (in sunset or sunrise), unexpected sudden illumination changes (shadow of clouds or rain) and high traffic density have unwanted effects on background.

The simplest background modeling method is running average that needs the lowest memory space. Because of mentioned weaknesses in running average method, few VDSs use running average method as basic algorithm for vehicle detection. However, some researches such as [15] were done to improve performance of running average.

Our proposed algorithm for vehicle detection is based on trip-line (trip-wire) approach. Trip-line approach is the simulation of loop detection using image processing methods for vehicle detection [11,15,16]. In trip-line approach, some rectangular regions are selected on image as detection regions. Therefore, only detection regions are investigated to find vehicle in each frame. Trip-line approach has lower computational complexity with respect to other approach that process the entire image frame. Usually, detection region is a rectangle that its size is equal with size of vehicle image.

In classic trip-line method, after background subtraction, if the ratio of foreground (detected) pixels number to total pixels number in a given region is more than a threshold (th_l), the region will be detected as a region that contains a vehicle. th_l is always between 0 to 1.

Our proposed algorithm uses fuzzy running average and fuzzy background subtraction. To determine vehicle existence in a detection region, mean value of fuzzy background subtraction matrix is calculated. If the mean value is more than a threshold (th_l), there will be a vehicle in the region.

5. Experimental Results

To compare the fuzzy running average and fuzzy background subtraction with classic running average and classic background subtraction, the application of these methods in vehicle detection is used. In experiments, a gray level camera is used for image acquisition. The output images are 640x480 pixels. Fig. 2 shows location of detection regions on a sample frame.

Duration of experiment was more than 2 hours in evening. Long time experiments let us to have more fair comparison. In addition, evening is the best time for test VDSs, because of extensive illumination changes and high-density traffic in freeway.

Table 2: An example of using standard background subtraction and fuzzy running average method.

time (frame #)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Input gray level	100	129	129	129	100	100	100	134	134	134	100	100	100	139	139
background value	100	100	100	100.1	100.1	100.1	100.1	100.1	100.1	100.1	100.1	100.1	100.1	100.1	100.1
detection status	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1
time (frame #)	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Input gray level	139	100	100	100	82	82	82	100	100	100	100	100	80	80	80
background value	100.1	100.1	100.1	100.1	100	99.9	99.8	99.8	99.8	99.8	99.9	99.9	99.9	99.8	99.7
detection status	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In classic running average and background subtraction, value of thresholds and parameters were set to values shown in Table 3. These have been selected by trial and error.

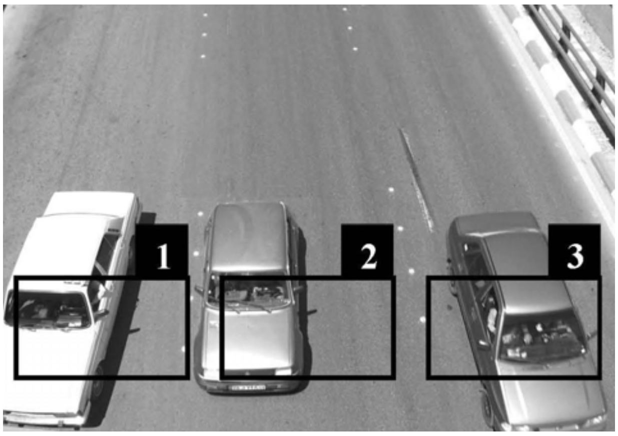

Fig. 2 Location of detection region on a sample frame.

Table 3: Thresholds and parameters used in classic vehicle detection

Phase	Background Subtraction	Background Updating		Vehicle Detection in Region
Threshold	th_s	α	th_u	th_l
Value	30	0.9999	20	0.4

To evaluate vehicle detection system, we used False Detection Rate (FDR) and False Rejection Rate (FRR). FDR shows the false detection error rate that system detects a vehicle in an empty region. FRR shows the false rejection error rate that system does not detect vehicle in an occupied region. Total error rate of VDS is summation of FDR and FRR. The ideal VDS must have 0% FDR and 0% FRR. Usually, vehicle's shadows and shadow of objects around the road increase the FDR. Moreover, vehicles with gray level similar to road gray level increase FRR.

Generally, in real systems, decreasing the FDR causes to increase FRR and vice versa. In trip-line based VDSs, increasing the th_s , α or th_l decreases FDR and increases FRR. Therefore, th_s , α and th_l were selected to minimize both FDR and FRR by trial and error. FDR and FRR are *4%* and 19% respectively. Consequently, total error rate is *23%*. The main reason of high FRR is that the illumination is reduced continuously in evening. Therefore, discrimination between vehicles and background is decreased. Fig. 3 shows result of vehicle detection on a sample frame. As shown in Fig. 3, classic running average and classic background subtraction could not detect vehicles in region 2 and 3.

Fig. 3 Result of vehicle detection using classic running average and classic background subtraction.

Classic approach for vehicle detection, only detected the vehicle in first region, because contrast of vehicle and background is very high. But, in region 2 and 3, gray level of vehicle is very similar to background gray level. So, vehicles in region 2 and 3 have not been detected.

Thresholds and parameter for fuzzy vehicle detection were selected similar to classic vehicle detection (Table 4). In background updating, α is computed based on equation (6) using a look-up table.

Table 4: Thresholds and parameters used in fuzzy vehicle detection

Phase	Background Subtraction	Background Updating	Vehicle Detection in Region
Threshold	th_s	$\alpha_{\min}$	th_l
Value	30	0.9	0.4

Experiments show that FDR and FRR of fuzzy vehicle detection are 5% and 12% respectively. It shows total error rate of fuzzy VDS is 6% lower than classic VDS with similar parameters. Fig. 4 shows result of fuzzy background subtraction on a sample frame. However, vehicle in region 3 could not be detected, but fuzzy approach could detect vehicles in region 1 and 2.

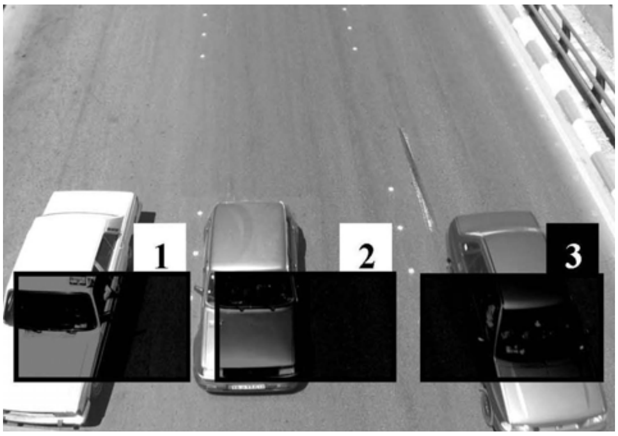

Fig. 4 Result of vehicle detection using fuzzy running average and fuzzy background subtraction.

Two algorithms were implemented in Microsoft C# .Net 2005 and were run on an AMD Athlon ™ 2800+ (1.8 GHz) with 512 MB RAM. Table 5 shows comparison between vehicle detection and fuzzy vehicle detection. Total error rate of vehicle detection and fuzzy vehicle detection are 23% and 17% respectively. However, processing rate of fuzzy vehicle detection is 22 frames per second (fps) and is about 12% slower than vehicle detection.

Table 5: Accuracy and complexity comparison between vehicle detection and fuzzy vehicle detection

	FDR	FRR	Error Rate	Frame Rate
Vehicle Detection	4%	19%	23%	25 fps
Fuzzy Vehicle Detection	5%	12%	17%	22 fps

6. Conclusions

In this study, some weaknesses of classic running average method for background modeling and background subtraction were mentioned in both an artificial and a real world example. In addition, a fuzzy approach for background modeling and background subtraction were proposed. For fuzzy background modeling, fuzzy running average was suggested.

To compare running average method and background subtraction with their fuzzy approaches in real world, both classic and fuzzy algorithms were used in vehicle detection application. Experiments have been done in evening, because of extensive illumination changes and high vehicle traffic density. Experimental results show that fuzzy approach is 6% more accurate than classic approach. However, fuzzy vehicle detection is 12% slower than classic vehicle detection.

Acknowledgments

First author wants to thank Mr. Mohamad Hadi Sigari, for his beneficial cooperation in experimental test. This project is partially supported by Bina Pardaz Shargh LTD, Mashad, Iran.

References

[1] B. P. L. Lo and S. A. Velastin, "Automatic congestion detection system for underground platforms", International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 158-161, 2000.
[2] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1337-1342, 2003.
[3] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 831-843, 2000.
[4] C. Stauffer, W. E. L. Grimson, "Adaptive background mixture modelsfor real-time tracking", IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 246-252, 1999.
[5] W. T. Dong, Y. S. Soh, "Image-based fraud detection in automatic teller machine", International Journal of Computer Science and Network Security (IJCSNS), vol. 6, no. 11, November 2006.
[6] A. H. S. Lai, N. H. C. Yung, "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence", IEEE International Symposium on Circuits and Systems, vol. 4, pp. 241-244, 1998.
[7] W. Wang, D. Chen, W. Gao, J. Yang, "Modeling background from compressed video", Second Joint IEEE International Workshop on Video Surveillance and Performance Evaluation of Tracking and Surveillance, in conjunction with the Tenth IEEE International Conference on Computer Vision. Beijing, China, October 2005.
[8] G. Christogiannopoulos, P. B. Birch, R. C. D. Young, C. R. Young, "Segmentation of moving objects from cluttered background scenes using a running average model". SPIE Journal, vol 5822, pp. 13-20. 2005.
[9] M. Yu, Y. D. Kim, "Vision based vehicle detection and traffic parameter extraction", IEICE Transaction on Fundamental Electronic, Communication and Computer Science, vol. E84-A, no. 6, pp. 1461-1470, 2001.
[10] O. Achler, M. M Trivedi, "Camera based vehicle detection, tracking, and wheel baseline estimation approach", the 7th International IEEE Conference on Intelligent Transportation Systems, pp. 743-748, October 2004.
[11] B. Coifman, D. Beymer, P. McLauchlan, J. Malik. "A real-time computer vision system for vehicle tracking and traffic surveillance", Trasportation Research: Part C, vol. 6, no. 4, pp. 271-288, 1998.

[12] L. Peihong, D. Liya, L. Jilin, "A video-based traffic information extraction system", IEEE Intelligent Vehicles Symposium, pp. 528-532, June 2003.
[13] P. S. Yen, C. Y. Fang, S. W. Chen, "Motion analysis of nearby vehicles on a freeway". IEEE International Conference on Networking, Sensing and Control, vol. 2, pp. 903- 908. 2004.
[14] A. Behrad, A. Shahrokni, S. A. Shahrokni, K. Madani, "A robust vision-based moving target detection and tracking system", Image and Vision Computing conference (IVCNZ2001). New Zealand. November 2001.
[15] Y. J. Kim, Y. S. Soh, "Improvement of background update method for image detector". 5th World Congress on Intelligent Transport Systems. 1998.
[16] M. H. Sigari, H. R. Pourreza, "An illumination invariant vehicle detection system". 1st National Congress on Computer, Electronic and IT Engineering. Hamedan, Iran. 2008.

Mohamad Hoseyn Sigari received the B.S. degree in Computer Engineering from Ferdowsi University of Mashad, Iran, in 2006 (with Honor). At present, he is M.S. Student in Department of Computer Engineering, in Iran University of Science and Technology, Iran. His research interests are Machine Vision, Image Processing, Intelligent Signal Processing, Soft Computing and Intelligent Transportation System.

Naser Mozayani received the B.S. degree in Computer Engineering from Sharif University of Technology, Iran, in 1990. He received the M.S. degree in Telematics & Information Systems from SUPELEC, France, in 1994 and the Ph.D. degree in Informatics from University of Rennes I, France, in 1998.
Currently, he is assistant professor in Department of Computer Engineering, Iran University of Science and Technology, since 1998. His research interests are Artificial Neural Networks, Fuzzy Logic, Handwritten Character Recognition and IT Applications.

Hamid Reza Pourreza received the B.S. degree in Electrical Engineering from Ferdowsi University of Mashad, Iran, in 1989. He received the M.S. and Ph.D. degrees in Electrical Engineering and Computer Engineering from Amirkabir University of Technology, Iran, in 1992 and 2003, respectively.
Now he is assistant professor in Department of Computer Engineering, Ferdowsi University of Mashad, Iran. His research interests are Machine Vision, Signal Processing, Hardware Design and Intelligent Transportation System.