A Genetic Algorithm-Based Moving Object Detection for Real-time Traffic Surveillance

Giyoung Lee, Rammohan Mallipeddi, Gil-Jin Jang, *Member, IEEE*, and Minho Lee

***Abstract*—Recent developments in vision systems such as distributed smart cameras have encouraged researchers to develop advanced computer vision applications suitable to embedded platforms. In the embedded surveillance system, where memory and computing resources are limited, simple and efficient computer vision algorithms are required. In this letter, we present a moving object detection method for real-time traffic surveillance applications. The proposed method is a combination of a genetic dynamic saliency map (GDSM), which is an improved version of dynamic saliency map (DSM) and background subtraction. The experimental results show the effectiveness of the proposed method in detecting moving objects.**

***Index Terms*—Background subtraction, dynamic saliency map, genetic algorithm, object detection, real-time traffic surveillance system.**

I. Introduction

MOVING object detection, a basic step in video analysis, is crucial in application areas such as automated visual surveillance, human-computer interaction, content-based video compression, and automatic traffic monitoring [1]. In other words, every tracking algorithm requires an object detection mechanism to detect objects either in every frame or during the first occurrence in the video [2], and the tracking performance of visual surveillance systems is dependent on the effectiveness of object detection. Moving object detection extracts moving objects of interest such as vehicles and pedestrians in video sequences with a static or dynamic background. In real-time traffic surveillance systems, moving object detection based on images obtained from fixed CCTV cameras involves many challenging problems including the following: 1) unexpected number of multiple moving objects; 2) size variation and poorly textured objects; 3) rapid change in illumination conditions; and 4) shadows and multiple occlusions [3].

In real-time embedded systems where the computational and memory resources are scarce, the computational complexity of different algorithms such as object detection is also a challenging task [4]. Therefore, object detection algorithms intended for embedded systems should be fast, simple and effective with superior performance.

In this letter, we present an object detection algorithm suitable to embedded surveillance applications. The proposed method is a combination of genetic dynamic saliency map (GDSM) and background subtraction. GDSM is based on dynamic saliency map (DSM) [3], and requires less computation, has higher object detection accuracy, and is more robust to noise and environment variations. The performance improvement of GDSM over the conventional DSM is due to optimization of the weights using a Genetic algorithm, while uniform weights are used in the original DSM when creating a saliency map. However, GDSM fails to detect objects especially when they stop unexpectedly in the middle of the road. Therefore, we combine GDSM with background subtraction (BS) to detect moving objects more accurately. BS helps detect tighter object boundaries and surrounding areas. Among the various BS methods [5], we adopt running Gaussian averaging that is known to be fast and simple.

The rest of the letter is organized as follows: Section II presents a brief literature review on different object detection algorithms. Section III describes the proposed algorithm in detail. Section IV presents the experimental results and discussion, and Section V concludes the letter.

II. Literature Review

There are many algorithms to detect moving objects with a fixed camera, such as the running Gaussian average, mixture of Gaussians, kernel density estimation (KDE), Eigen backgrounds, Markov random field (MRF), dynamic conditional random field (DCRF) [1], [2], [5], [6]. Recently, background subtraction (BS) [1], [5], [7], [8] has been widely adopted in moving object detection, which classifies each pixel in the scene as foreground, shadow, or background. The object detection algorithm is generally composed of the following four steps: 1) foreground detection, 2) pixel level post-processing, 3) detecting connected regions, and 4) region level post-processing [2].

In order to extract the foreground object, BS calculates the difference between the current image and the reference (background) image. In the case of the running Gaussian average, the background image is assumed to have no moving objects and regularly updated to adapt to the background changes. However, mixture of Gaussians focuses on robust background modeling and updating to adapt the background model to the varying illumination conditions during different times of the day, geometry reconfiguration of the background structure, and repetitive motion from clutter.

Manuscript received October 31, 2014; accepted March 20, 2015. Date of publication March 30, 2015; date of current version April 02, 2015. This work was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (NIPA-2014-H0401-14-1004) supervised by the NIPA(National IT Industry Promotion Agency). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Negar Kiyavash.

The authors are with the School of Electronics Engineering, Kyungpook National University, Taegu, Korea (e-mail: giyoung0606@gmail.com; mallipeddi.ram@gmail.com; gjang@knu.ac.kr; mholee@knu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2417592

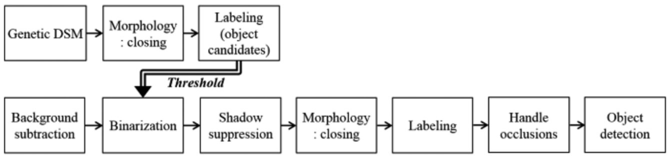

Fig. 1. Outline of proposed object detection algorithm.

Unlike the aforementioned algorithms that utilize the information from a single frame in a video sequence, some other object detection techniques make use of the temporal information from a sequence of consecutive frames to reduce the number of false detections. Temporal differencing [9] uses the pixel-wise differences between two or three consecutive images in an image sequence to extract moving regions and is highly adaptive to the dynamic scene changes. However, algorithms based on temporal difference fail to extract all the relevant pixels of a foreground object especially when the object has uniform texture or moves slowly [9].

In addition to the temporal information, spatial information and contextual constraint are important to handle non-stationary background variations and understand the complex structure of the scene. Algorithms such as MRF, DCRF and Wang's method [1] consider both temporal and spatial information for reliable object detection. In addition, in Wang's method [1], the cast shadow is removed from the foreground objects. Self-Balanced SENsitivity SEgmenter (SuBSENSE) [8] also use spatio-temporal feature descriptor to increase spatio-temporal sensitivity. SuBSENSE use both local color intensity and spatio-temporal neighborhood similarity. SuBSENSE also focuses on balancing the inner workings of a non-parametric model based on pixel-level feedback loops. Therefore, SuBSENSE can make an adaptive and flexible background.

DSM [3], unlike BS, is not based on background model so it can overcome the drawbacks of BS. DSM is used to analyze the dynamics of the successive static saliency maps, and can localize an attention region in dynamic scenes to focus on a specific moving object for traffic surveillance. In DSM, the entropy maximum through time is used to analyze the dynamic characteristics of successive static saliency maps.

III. The Proposed Method

The proposed method combines the GDSM and BS to detect moving objects from images captured by a fixed camera mounted on a streetlight. Fig. 1 illustrates the proposed object detection procedure by a block diagram.

As shown in Fig. 1, GDSM is performed first and blobs are detected. After isolating the blobs in the GDSM using morphological closing with a 3×3 rectangularly structured element, the moving object candidates are localized using the grassfire labeling method. Then, background subtraction is carried out using the threshold obtained from the object candidates using the GDSM. In the binarization process, the shadow pixels are removed from the foreground pixels. If an object occlusion is detected, the partially occluded objects are searched again.

A. Genetic Dynamic Saliency Map (GDSM)

GDSM is an improvement of the DSM model [3] with the help of a genetic algorithm (GA) [10]. The outline of the GDSM is depicted in Fig. 2. In addition, the dynamic analysis in DSM involves a lot of computation due to entropy calculation. In the proposed method, we alleviate the time complexity by using a fast dynamics analysis where entropy is replaced with variance.

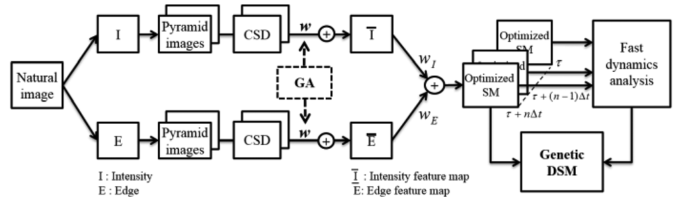

Fig. 2. Outline of the GDSM.

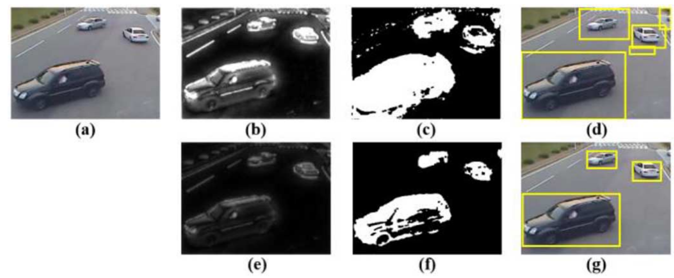

Fig. 3. Comparison of the results obtained using uniform weights (b), (c), (d) and optimized weights (e), (f), (g) of CSD using GA; (a) input image, (b&e) SM, (c&f) DSM, and (d&g) object detection result.

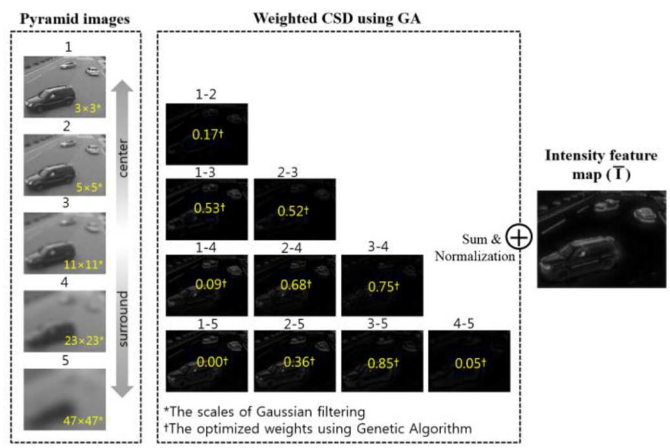

Fig. 4. Optimized weights of CSD (intensity part) using GA.

Optimizing Saliency Map Using Genetic Algorithm: In the process of creating the saliency map (SM), weighted center-surround difference (CSD) is used to reduce blur noise. The blur noise around objects, generated by the repetitive image resizing in the Gaussian Pyramid, may reduce the object detection performance. As shown in Fig. 3(d), the sizes of the detected objects are bigger than the real sizes. In this letter, the weights of CSD are optimized using GA, resulting in tighter object regions in Fig. 3(g).

GA starts with a population of randomly initialized weights that are being optimized with the lower and upper bounds, which are zero and one, respectively. The average error of object detection in all training images is calculated based on the overlap ratio, and used in calculating the fitness function defined as in (1). The overlap ratio is calculated by the ratio of the intersection area to the union area of the hand-labeled ground truth and the detected object area by the algorithm:

$$fitness\ function = E(1 - overlap\, ratio) \quad (1)$$

$$overlap\ ratio = \frac{A_{T\cap D}}{A_{T\cup D}} = \frac{A_C}{A_T + A_D - A_C} \quad (2)$$

TABLE I
PARAMETERS USED IN GA

Maximum generation	Population size	Selection rate	Crossover rate	Mutation rate	Chromosome length
100	20	0.9	0.8	0.2	20

where A_T is the area of the ground truth of the target object, and A_D is the area of the detected object using GDSM. A_C is the area of the intersection. The GA algorithm adapts the system parameters to minimize the fitness function in (1), leading to the accurate and tight detection by maximizing the overlap ratio in (2). Table I shows the parameters of GA [10].

Fast Dynamics Analysis: In DSM [3], maximum entropy among the different consecutive frames over a given period is used to analyze the dynamic characteristics of successive SMs. The entropy is calculated from the histogram of pixel values corresponding to the local region. Then, the DSM is binarized using an empirical threshold. However, the entropy calculation is too complex and is computationally expensive rendering it unsuitable for embedded and real-time systems. Therefore, in this letter, the variance calculated from pixel values corresponding to the local region is used to analyze the dynamic characteristics of successive SMs instead of entropy. If the variance is larger than given threshold ($= 6$), the pixel is considered as foreground.

B. Combination of GDSM and BS

After obtaining GDSM, the background image is calculated with a running Gaussian average as follows:

$$\mu_t(x,y) = \begin{cases} \mu_{t-1} & , if\ x,y \in O_{GDSM} \\ \alpha I_t + (1-\alpha)\mu_{t-1} & , otherwise \end{cases} \quad (3)$$

where I_t is the value of the current pixel, and μ_t and μ_{t-1} are the average values of the current and previous pixels, respectively. α is an empirical weight ($\alpha = 0.2$). To increase accuracy, background is updated except moving object candidates O_{GDSM} which is detected by GDSM.

Then, at each time frame t, the pixel (*x,y*) is classified as a foreground if

$$|I_t(x,y) - \mu_t(x,y)| > TH(x,y) \quad (4)$$

where the threshold, $TH(x,y)$, is obtained from the candidate foreground region O_{GDSM}.

$$TH(x,y) = \begin{cases} \max(|E(I_t) - E(\mu_t)|, TH_{\min}) & \text{if}\, x,y \in O_{GDSM}, \\ TH_{\max}, & \text{otherwise} \end{cases} \quad (5)$$

where $E(I_t)$ and $E(\mu_t)$ are the mean values of the current and background pixels, respectively, in O_{GDSM}. TH_{min} and TH_{max} are the minimum and maximum threshold values ($TH_{min} = 20, TH_{max} = 120$).

C. Shadow Suppression

The shadows are removed with a simple method proposed in [11]. However, the shadows are removed only with the intensity instead of the HSV color information to reduce the computational load. In each pixel of an input image, the shadow is detected by (6). $I(x,y)$ is the pixel value of the input image and μ_{back} is the average pixel value of the background image which is generated by BS. β_{low} and β_{high} are the constant threshold

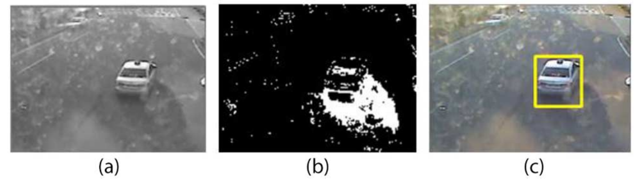

Fig. 5. The result of shadow suppression. (a) The input image (b) Shadow detection (c) Foreground segmentation without shadow.

Fig. 6. The result of re-detection of occluded objects. The middle image shows the re-detected objects in the case of occlusions with the mean-shift algorithm.

($\beta_{low} = 0.7$, $\beta_{high} = 0.9$). The pixels corresponding to the shadow are removed from the foreground pixels.

$$SP(x,y) = \begin{cases} 1 & , \text{if}\, \beta_{low} \le I_t(x,y)\mu_{back}^{-1} \le \beta_{high} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

D. Dealing with Partial Occlusions

If one object occlude the other, the proposed GDSM and BS may detect the two object as a single object. In this letter, occlusions are detected and dealt with after detecting moving objects by BS. The detected region of the k-th object in the current frame is denoted by O_t^k and the center position of the i-th object in the previous frame is denoted by C_{t-1}^i. The k-th detected object is considered as being occluded if

$$\left|\{i | C_{t-1}^i \in O_t^k\}\right| \ge 2 \quad (7)$$

where the cardinality operator $|\cdot|$ returns the number of elements in the set. If the k-th detected object is occluded by i-th object, it is searched with the mean-shift algorithm [12] within O_t^k.

IV. EXPERIMENTAL RESULTS

The proposed method was evaluated by the detection performance of moving objects in video clips, captured by fixed CCTV cameras whose resolution is 160×120. The experiments were conducted on an Intel i7 Quad Core 3.50 GHz PC with 16G RAM installed. To evaluate the performance of the proposed method under different weather conditions, three different data sets were employed. Data set 1 is collected under cloudy weather, data set 2 under rainy weather, and data set 3 in sunny days, with shadows. In the case of data set 3, due to the stains on the camera by the rain, some detection performance degradation is observed. Fig. 5 shows the result of the shadow suppression, showing that the shadows can be effectively detected and removed by the proposed method. Although some pixels are mistakenly detected as shadows, the proposed method is successful in detecting the object region, as well as occlusions in each frame. Once the occlusion is detected, the occluded objects are searched with the mean-shift algorithm as shown in Fig. 6. Therefore, the proposed method can detect and identify multiple objects unlike the conventional methods that identify all the occluded objects as a single object.

TABLE II
PERFORMANCE OF DIFFERENT OBJECT DETECTION METHODS

	Condition	# Frames	# Objects	Detection Accuracy [%]				False Alarm Rate [%]			
				DSM[3]	BS[5]	SuBSENSE[8]	Proposed	DSM[3]	BS[5]	SuBSENSE[8]	Proposed
Data Set 1	Cloudy	10212	5747	84.76	94.24	97.15	97.69	3.54	3.57	0.05	0.30
Data Set 2	Rainy	10628	6410	76.79	93.95	94.38	94.13	3.79	3.59	0.13	0.70
Data Set 3	Sunny	10110	3658	88.96	93.44	94.64	94.72	2.74	5.53	0.23	1.15

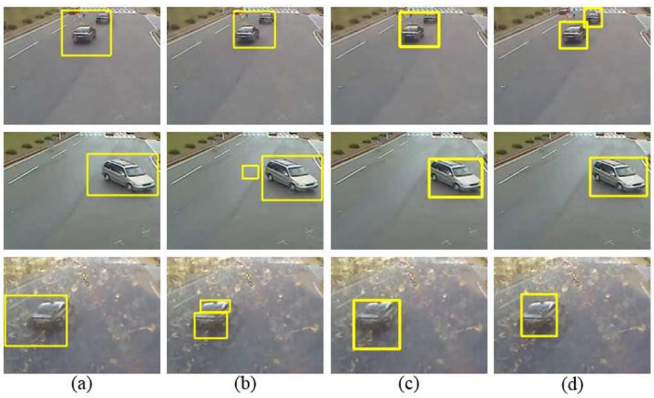

Fig. 7. Images showing object detection by different detection algorithms. The first low is cloudy condition, second low is rainy condition, and third low is sunny condition. (a) DSM (b) BS (c) SubSENSE (d) proposed method.

Table II shows the detection performances of the different object detection algorithms under different conditions by using detection accuracy and false alarm rate as follows:

$$Detection\ Accuracy = \frac{\#correctly\ detected\ objects}{\#total\ objects} \times 100 \quad (8)$$

$$False\ Alarm\ Rate = \frac{\#false\ detected\ objects}{\#total\ frames} \times 100 \quad (9)$$

The detection accuracy was measured at the object level. A detected object using a detection algorithm is regarded as a success if the overlap ratio, calculated by (2), is larger than 0.5 and as a false detection if the overlap ratio is zero.

As shown in Fig. 7, DSM [3] detects areas that include the entire object of interest, although the size of the identified area is much larger than the object itself. Unlike DSM, the area detected by BS [5] is more compact, but it misses some parts of the objects. However, SuBSENSE [8] and the proposed method tightly detect objects without missing any parts of the objects. In addition, SuBSENSE and the proposed method are robust to environmental conditions as well as more accurate compared to the DSM and BS. However, as shown in Table III, the average time taken by SuBSENSE is too much and is not suitable for embedded systems. The proposed method needs less execution time than the combined execution time of the DSM and BS because of the reduced computational load of the GDSM compared to the DSM. The improved detection accuracy results are mainly from the optimized weights of CSD in the GDSM algorithm that is obtained by the GA. Therefore, the GDSM is faster and has higher accuracy than original DSM as shown in Table IV.

TABLE III
AVERAGE PROCESSING TIME [MSEC]

DSM [3]	BS [5]	SuBSENSE [8]	Proposed
30.0	8.4	290.9	23.4

TABLE IV
PERFORMANCE OF GDSM (DATA SET 1)

	Detection Accuracy [%]	False Alarm Rate [%]	Processing Time [msec]
DSM [3]	84.76	3.54	30.0
Genetic DSM	91.46	5.52	17.3

V. CONCLUSION

In this letter, we propose an improved object detection method based on a genetic dynamic saliency map and background subtraction. The proposed object detection method is fast, simple, and with good performance. Thus, it is suitable for embedded systems. The proposed method is practical to handle shadow and occlusion problems efficiently.

REFERENCES

[1] Y. Wang, “Real-time moving vehicle detection with cast shadow removal in video based on conditional random field,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 437–441, 2009.
[2] K. A. Joshi and D. G. Thakore, “A survey on moving object detection and tracking in video surveillance system,” *Int. J. Soft Comput. Eng. (IJSCE)*, vol. 2, no. 3, pp. 44–48, 2012.
[3] J.-W. Woo, W. Lee, and M. Lee, “A traffic surveillance system using dynamic saliency map and svm boosting,” *Int. J. Contr. Auton. Syst.*, vol. 8, no. 5, pp. 948–956, 2010.
[4] B. Rinner and W. Wolf, “An introduction to distributed smart cameras,” *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, 2008.
[5] M. Piccardi, “Background subtraction techniques: A review,” in *IEEE Int. Conf. Syst., Man and Cybern*, 2004, vol. 4, pp. 3099–3104.
[6] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong, “Intelligent visual surveillance—A survey,” *Int. J. Contr. Auton. Syst.*, vol. 8, no. 5, pp. 926–939, 2010.
[7] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, 2003.
[8] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “Flexible background subtraction with self-balanced local sensitivity,” in *IEEE Conf. Comput. Vision and Pattern Recogn. Workshop (CVPRW)*, 2014, pp. 414–419.
[9] N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 3, pp. 266–280, 2000.
[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addion-Wesley, 1989.
[11] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, “Improving shadow suppression in moving object detection with HSV color information,” in *Proc. IEEE Intell. Transp. Syst.*, 2001, pp. 334–339.
[12] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of nonrigid objects using mean shift,” in *IEEE Conf. Comput. Vision Pattern Recognition*, 2000, pp. 142–149.