British Journal of Applied Science & Technology
4(29): 4148-4155, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Genetic Algorithm for Optimizing Background Subtraction Parameters in Computer Vision

Ramesh Rajagopalan[1*]

[1]*School of Engineering, University of St. Thomas, St. Paul, MN, USA.*

Author's contribution

The sole author designed, analyzed and interprets and prepared the manuscript.

Original Research Article

Received 17th June 2014
Accepted 19th July 2014
Published 7th August 2014

ABSTRACT

Tracking moving objects in a video sequence is a critical task in several computer vision applications. A common approach is to perform background subtraction which identifies moving objects in a video frame. The mixture of Gaussians model is one of the most popular techniques for performing background subtraction. The performance of the mixture of Gaussian model strongly depends on parameters such as learning rate, background ratio, and number of Gaussians. Fine tuning these parameters is a huge challenge for efficient performance of the background subtraction algorithm. In this work, we propose a genetic algorithm to determine the optimal values of the learning rate and background ratio. Experiments based on the Wallflower test images demonstrate the superior performance of the genetic algorithm when compared to a recently proposed particle swarm optimization approach.

Keywords: Mixture of gaussians; background subtraction; genetic algorithm.

1. INTRODUCTION

With the arrival of fast and efficient computer programs working with real-time data, image processing has exploded into many vast fields. One such field is video surveillance. In this paper, we consider the problem of video surveillance and monitoring. An efficient surveillance system must be capable of handling lighting changes, cluttered background,

**Corresponding author: E-mail: ramesh@stthomas.edu;*

shadows, and moving objects. Background subtraction is the very first step in several computer vision applications. The background subtraction algorithm should be able to accurately extract the foreground pixels corresponding to the moving object. Recently, researchers have used adaptive backgrounding for effective tracking of moving objects where the images are averaged over time with a predetermined threshold for the entire scene [1,2]. Several background subtraction techniques have been proposed in the literature [3-8]. Among these, the mixture of Gaussians (MoG) model proposed by Staufer and Grimson is widely used due to its robustness to variations in lighting and higher accuracy [3]. However, the performance of this approach is dependent upon choosing the appropriate values of parameters such as learning rate and background ratio. The ideal values of these parameters change for different scenarios such as indoor vs. outdoor. Fine tuning these parameters poses significant challenges on the adaptability of the approach for a different applications. Due to these issues, end users have significant challenges choosing the right parameter values for obtaining optimal results using the background subtraction algorithm.

To address these challenges, we propose an optimization technique using a genetic algorithm to determine the optimal values of parameters such as learning rate and background ratio. This will eliminate the need for manual fine tuning of the parameters and provide the ability to perform efficient background subtraction for a variety of scenarios. We develop a fitness function that determines the similarity of the results obtained from the background subtraction technique to the ground truth image. We compare the performance of the genetic algorithm with a particle swarm optimization (PSO) approach proposed in [9]. Our simulation results show that the genetic algorithm outperforms PSO on a variety of test images in the wallflower data set.

2. MIXTURE OF GAUSSIANS

In this section, we describe the mixture of Gaussians model proposed in [3]. Background modeling is a key element of a background subtraction algorithm. Several researchers have developed background modeling techniques for identifying moving objects of interest. Among these, the mixture of Gaussians approach is widely popular because of its high accuracy and ability to handle multimodal background distributions. The MoG model maintains a probability density function for each pixel. The pixel distribution $f(I_t = u)$ is modeled as a mixture of *K* Gaussians

$$f(I_t = u) = \sum_{i=1}^{K} w_{i,t}\,\eta(u; \mu_{i,t}, \sigma_{i,t}) \tag{1}$$

Where I_t denotes the luminance pixel intensity at time *t*, $\eta(u; \mu_{i,t}, \sigma_{i,t})$ is the i^th^ Gaussian distribution with mean $\mu_{i,t}$, standard deviation $\sigma_{i,t}$ and $w_{i,t}$ is the proportion of data accounted for by the i^{th} component. The parameter *K* indicates the total number of Gaussian distributions. The weights $w_{i,t}$ are updated as

$$w_{i,t} = (1-\alpha)w_{i,t-1} + \alpha \tag{2}$$

where α is the learning rate and $0 \le \alpha \le 1$. The Gaussians are ordered by the value of $w_{i,t} / \sigma_{i,t}$. After sorting, the first M components that satisfy the following criteria are declared to be the background components.

$$\sum_{j=1}^{M} w_{j,t} \ge T \tag{3}$$

Where *T* is a measure of the minimum portion of data that should be accounted for by the background. The learning parameter and background ratio are crucial parameters of the MoG model. The learning rate governs how rapidly the algorithm adapts to changes in a scene. For simple scenarios a small value of learning rate will enable adaptation to illumination changes and other minor modifications in the background. However in more complex scenarios such as an outdoor setting, a higher learning rate might be neededto accommodate rapid changes in illumination and factors such as wind and movement of trees. The background ratio specifies the probability of a pixel value belonging to the background. If the value of *T* is very low, only some of the modes might be considered background. A large value of *T* may cause foreground distributions to represent the background. Hence, choosing optimal values of learning rate and background ratio is a very challenging task and needs fine tuning to suit the specific application at hand. In the next section, we describe a genetic algorithm for finding the optimal values of these parameters.

3. METHODOLOGY

Our goal is to determine the vector x= {*α*, *T*} that maximizes the following objective functions:

- Recall: $f_1(x)$
- Precision: $f_2(x)$

where

$$f_1(x) = \frac{\text{Number of foreground pixels correctly identified by the algorithm}}{\text{Number of foreground pixels in the ground truth}}$$

$$f_2(x) = \frac{\text{Number of foreground pixles correctly identified by the algorithm}}{\text{Number of foreground pixels detected by the algorithm}}$$

Maximization of precision reduces the percentage of false positives while maximizing the recall reduces the number of false negatives. There is a tradeoff between recall and precision. Recall increases with the number of foreground pixels detected which results in a decrease in precision. The classical approach to solve a multi-objective optimization problem is to assign a weight to each objective function and optimize the resulting single objective function. Hence, we define a new objective function $z(x)$ as

$$z(x) = \omega_1 f_1(x) + \omega_2 f_2(x) \tag{4}$$

Where ω_1 and ω_2 are the weights such that $\omega_1 \in (0,1)$, $\omega_2 \in (0,1)$ and $\omega_1 + \omega_2 = 1$

3.1 Weight Based Genetic Algorithm

Genetic algorithms are a family of computational models inspired by evolution [10]. They use a population of initial sample points in search space together with selection and recombination operators to generate new sample points. The goal of the genetic algorithm is to find the optimal values of learning rate and background ratio for maximizing the fitness function *z(x).* In this work, we propose a weight based genetic algorithm where each solution vector x_i in the population uses a different weight vector $\omega_i = \{\omega_1, \omega_2\}$. The weight vector is embedded within the solution vector x_i such that $x_i = \{\alpha_i, T_i, \omega_i\}$. Fig. 1 describes the various steps of the genetic algorithm.

1. Generate a random set of solutions x_i constituting a population P_t of size *n*
2. Assign a fitness value to each solution x_i by performing the following steps:

   2.1 Generate a random number $u_k \in [0,1]$ for each objective *k* where *k*=1,2.

   2.2 Calculate the random weight of each objective k as $\omega_k = (1/u_k)\sum_{i=1}^{2} u_i$

   2.3 Compute the fitness of each solution $z(x_i)$
3. Calculate the selection probability $p(i)$ of each solution x_i as
$$p(i) = \frac{z(x_i)}{\sum_{j=1}^{n} z(x_j)}$$
4. Select parents using the selection probabilities in step 3. Apply crossover and mutation operators to generate offspring. Replace the solutions in parent population P_t with the new off spring.
5. If the termination condition is not satisfied, move to step 2.

Fig. 1. Weight based genetic algorithm

We use a uniform crossover operator where each element (parameter) in the offspring is created by copying the corresponding element from one or other parent according to a random crossover mask. Each offspring undergoes a Gaussian mutation technique where a Gaussian distributed random value with mean zero and variance one is added to each element. Details of the crossover and mutation operators are discussed in [11,12]. The background subtraction algorithm poses limits on the values of learning rate and background ratio. Hence, if the mutation operator results in a parameter value outside the limits, its value is forced to the minimum or maximum of the corresponding parameter.

4. RESULTS AND DISCUSSION

We performed several simulations based on Wallflower test images [13]. The data base includes challenging scenes such as sudden illumination change, clutter motion, and slow moving foreground objects. The mixture of Gaussians model is used for all experiments. We set the number of Gaussians to five while varying the background ratio and the learning rate. We compare the performance of the weight based genetic algorithm (GA) with the particle swarm optimization (PSO) approach proposed in [9]. We use a population size of 30 for both GA and PSO and the number of generations was set to 40. Fig. 2 shows the variation of fitness function *z(x)*with the number of generations for the waving trees test image shown in Fig. 3a. We observe that the genetic algorithm outperforms PSO and converges within 20 generations. The GA converges to a fitness value of 0.87while compared to PSO which converges to 0.5. This also shows that the GA obtains better precision and recall by optimizing the learning rate and background ratio.

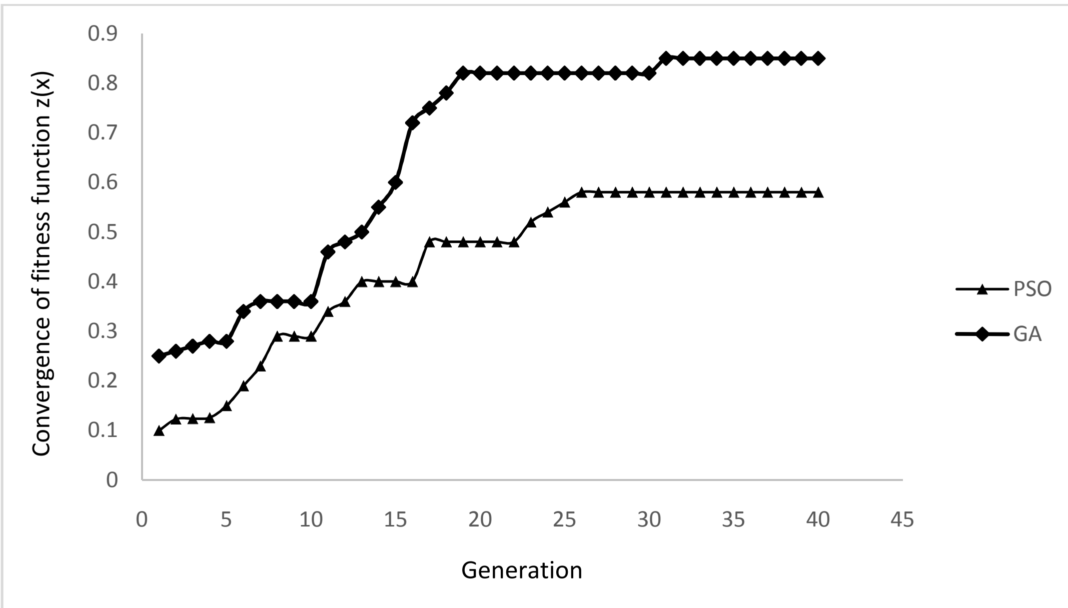

Fig. 2. Performance comparison of genetic algorithm and particle swarm optimization

Figs. 3a and 3b show the test image and the ground truth of a scenario obtained from the wall flower data set. The waving trees in the background pose a significant challenge in the background subtraction algorithm.

Fig. 3a. Test image of the waving trees scenario

Fig. 3b. Ground truth of the waving trees scenario

Figs. 4a and 4b show the results obtained by PSO and GA respectively. From these figures, we observe that the GA outperforms PSO and obtains a binary mask that closely resembles the test image. This shows that the genetic algorithm was able to obtain the optimal values of learning rate and background ratio while handling challenges such as waving trees in the background.

Fig. 4a. PSO results for waving tree scenario

Fig. 4b. GA results for waving trees scenario

Figs. 5a and 5b show the test image and ground truth of another scenario in the wall flower data set. In this image, a static foreground occludes the dynamic background. Figs. 6a and 6b show the results obtained by PSO and GA respectively. We observe that the genetic algorithm obtains a video mask that closely resembles the ground truth. This shows the superior performance of the GA when compared to PSO while handling occlusion.

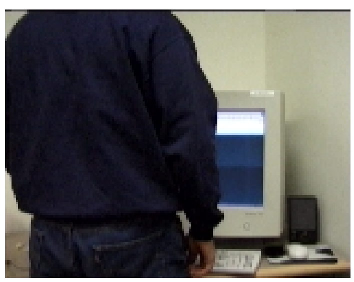

Fig. 5a. Test image of the foreground occlusion scenario

Fig. 5b. Ground truth of the foreground occlusion scenario

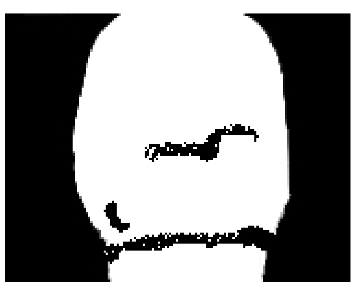

Fig. 6a. PSO results for the foreground occlusion scenario

Fig. 6b. GA results for the foreground occlusion scenario

We also performed experiements on several other test images from the wallflower data set. We observed that the gentic algorithm obtained much better results when compared to PSO on all test images.

To assess the effeciency of the genetic algorithm, we performed simulations for detecting a personin a room with lighting changes. Figs. 7a and 7b show the frames of the original video used in our simulations. Fig. 7a shows two people entering the room at t=3 seconds. There is a significant lighting change at t=4 seconds, which is the main source of background noise. Fig. 7b shows the person leaving the room at t=8 seconds. The total duration of the video is 9 seconds.

Fig. 7a. Original video at t=3 seconds

Fig. 7b. Original video att=8 seconds

Figs. 8a and 8b show the results obtained by GA and PSO respectively. We observe that the GA outperforms PSO by eliminating background noise completely. This demonstrates the ability of the genetic algorithm to perform efficient and robust tracking.

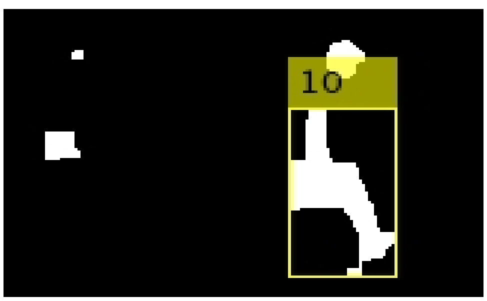

Fig. 8a. PSO results for the video at t=8 seconds

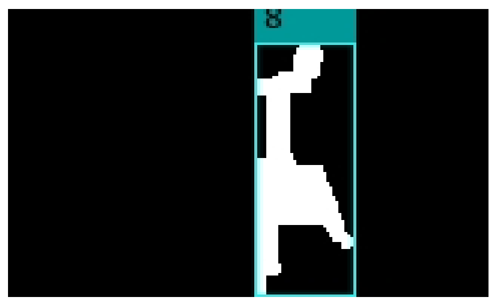

Fig. 8b. GA results for thevideo at t=8 seconds

5. CONCLUSION

This paper proposed a genetic algorithm for optimizing the learning rate and background ratio in the mixture of Gaussians model. Fine tuning these parameters is a complex task and varies for different applications. To alleviate this problem, we proposed a genetic algorithm for optimizing the parameters thus improving the performance of the background subtraction algorithm. We used two important performance metrics - precision and recall to develop a fitness function. We compared the performance of the genetic algorithm with a recently proposed particle swarm optimization approach. Simulations on several videos from the wall flower data set showed that the GA outperforms PSO and converges within twenty generations. Future work will investigate multi-objective optimization approaches for simultaneously maximizing the precision and recall. We also plan to investigate approaches for optimizing other parameters such as the number of Gaussians and the number of training frames to eliminate fine tuning of such parameters for different videos.

ACKNOWLEDGEMENTS

The author would like to thank his student Timothy Dessonville for providing assistance with the simulations.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Wren C, Azarbayejani A, Darrell T, Pentland AP. Pfinder: Real time tracking of the human body. IEEE Trans. On Pattern Anal and Machine Intell. 1997;19(7):780-785.
2. Cheng L, Gong M. Realtime background subtraction from dynamic scenes. Proc IEEE International Conference on Computer Vision; 2009.
3. Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 1999;246-252.
4. Ridder C, Munkelt O, Kirchner H. Adaptive background estimation and foreground detection using Kalman filtering. Proc International Conference on recent Advances in Mechtronics.1995;193-199.
5. Koller O, Weber J, Huang T, Malik J, Ogasawara G, Rao B, et al. Towards robust automatic traffic scene analysis in real time. Proc of the International Conference on Pattern Recognition; 1994.
6. Piccardi M. Background subtraction techniques: A review. In Proceedings IEEE International conference on systems, man and cybernetics; 2004.
7. Bouwmans T, El Baf F, Vachon B. Background modeling using mixture of Gaussians for foreground detection-a survey. Recent Patents on Computer Science. 2008;219-237.
8. Bamich O, Van Droogenbroeck M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing. 2011;20(6):1709-1724.
9. White B, Shah M. Automatically tuning background subtraction parameters using particle swarm optimization. Proc IEEE International Conference on Multimedia and Exposition; 2007.
10. Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison Wesley. Reading: Massachusetts; 1998.
11. Back T, Schwefel HP. An overview of evolutionary algorithms for parameter optimization. Evol Comput. 1993;1(1):1–23.
12. Schaffer DJ. Uniform crossover in genetic algorithms. Proc Third International Conference on Genetic Algorithms. 1989;2-9.
13. Toyoma K. Wallflower: Principles and practice of background maintenance. Accessed May 2104. Available: http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=621&id=5&aid=5658