

Moving Object Detection using Differential Evolution

Amlan Raychaudhuri, Arkadev Roy, Ashesh Das, Gourav Kumar Shaw, Pratik Kumar Mitra

Abstract—Moving Object detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. Different complex algorithms are employed to detect a moving object in a video. It has large number of applications in video surveillance and other security systems that are used to process video information. We have achieved it using Differential Evolution (DE). The proposed method is successfully tested over two video sequences.

Index Terms— Clustering, Differential Evolution, Moving Object Detection, Temporal video segmentation.

I. INTRODUCTION

The use of video has a very prominent role in today's surveillance systems. It is used to record the movement of traffic, provide real time information of remote locations such as offices, ATM counters, market place for proper monitoring. As the amount of information provided can be very large, it is desirable to have a system which is capable of detecting motion or changes in a video stream so that proper actions may be taken.

A video file is composed of a number of frames, each frame being a snapshot at a particular moment and by monitoring these frames with a particular base frame we may be able to understand whether motion has taken place during a particular time interval. However there are a number of challenges to this task of detecting motion in a video. We may need to take into account several issues like change in lighting conditions over a period of time, motion of leaves of trees in the background, etc.

Various methods have been proposed for Moving Object Detection such as the Linear Dependence and Vector Model and the Wronskian Model [7]. An extensive evaluation procedure is devised consisting of subjective [4] and objective parameters [5], [6] to compare these algorithms [1]-[3].

Different evolutionary techniques are used to detect a moving object in a video sequence. Some of them are applied for spatiotemporal spatial segmentation and some others are used for temporal video segmentation. Simulated Annealing [10], Genetic Algorithms [11]-[14] are used for spatiotemporal spatial and temporal both segmentation. In this paper, Differential Evolution technique is used to temporal segmentation of a video sequence.

Manuscript Received on May, 2014.

Amlan Raychaudhuri, Dept. of CSE, B. P. Poddar Institute of Management & Technology, Kolkata, India.

Arkadev Roy, Dept of CSE, B. P. Poddar Institute of Management & Technology, Kolkata, India.

Ashesh Das, Dept of CSE, B. P. Poddar Institute of Management & Technology, Kolkata, India.

Gourav Kumar Shaw, Dept of CSE, B. P. Poddar Institute of Management & Technology, Kolkata, India.

Pratik Kumar Mitra, Dept of CSE, B. P. Poddar Institute of Management & Technology, Kolkata, India.

Differential Evolution (DE) is a method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

Differential Evolution optimizes a function [8], [9] with D real parameters. At first we must select a population of size N (it must be at least 4) and the parameter vectors have the form:

$$x_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}], i = 1, 2, \dots, N$$

where G is the generation number,

and $x_{j,i,1} \in [x_{jl}, x_{jh}]$ where x_{jl} is a lower bound and x_{jh} is the upper bound. This is the first phase of the algorithm and is known as the initialization phase.

Each of the N parameter vectors then undergoes mutation, recombination and selection. Mutation expands the search space. For a given parameter vector $x_{i,G}$, three vectors $x_{r1,G}$, $x_{r2,G}$ and $x_{r3,G}$ are selected such that the indices i, r1, r2 and r3 are distinct. Then the weighted difference of the two vectors are added to the third using the following formula

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G})$$

where F is the mutation factor lying between 0 and 2. Vector $v_{i,G+1}$ is called the donor vector.

After mutation the next phase is recombination. Here, successful solutions from the previous generation are incorporated. The trial vector $u_{i,G+1}$ is developed from the elements of the target vector, $x_{i,G}$, and the elements of the donor vector, $v_{i,G+1}$. The elements of the donor vector enter the trial vector with probability C_R .

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } \text{rand}_{j,i} \leq CR \text{ or } j = I_{\text{rand}} \\ x_{j,i,G} & \text{if } \text{rand}_{j,i} > CR \text{ or } j \neq I_{\text{rand}} \end{cases} \quad (1)$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, D$

$\text{rand}_{j,i} \sim U[0, 1]$, I_{rand} is a random integer from [1, 2, ..., D] ensuring that $v_{i,G+1} \neq x_{i,G}$

The final stage of the algorithm is selection where the target vector $x_{i,G}$ is compared with the trial vector $v_{i,G+1}$ and the one with the lowest function value is admitted to the next generation.

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad i=1, 2, \dots, N \quad (2)$$

The stages mutation, recombination and selection continue until some stopping criterion is reached.

The organization of this paper is as follows. Section II describes the proposed methodology for moving object detection. Experimental results are given in Section III and conclusions are provided in Section IV.

II. PROPOSED METHOD

In this paper we have worked on gray level video sequences. Initially all the frames are extracted from the video sequence. Then we have chosen a reference frame, where no object is present.



And also we have taken a target frame, in which object will be detected. Now from these two frames by computing the absolute difference we have got the difference image frame. Then the proposed method is working on the difference frame image.

The paper consists of several modules each doing a particular task which is illustrated below:

A. Initialization - Storing and padding of the pixel values of the image

Our input file consists of $m \times n$ pixels intensity values which are kept in a .jpg file. Then it stores the intensity values specified for each pixel up to $m \times n$ pixels. These values are read from the file and stored in the array of size $m \times n$. The boundary values are padded by the copying the values of the extreme positions to generate a $(m+1) \times (n+1)$ matrix. Then we store nine digit (the pixel itself and its eight closest neighbors) patterns with respect to each pixel value (one pattern corresponding to each pixel of the original image).

Table I shows the pixel $x[i,j]$ with its surrounding pixels. Each pixel in the image is stored with its neighbors in an array like Table II and create a pattern for a particular pixel.

B. Chromosome formation

Two patterns are randomly chosen from the $m \times n$ patterns generated in earlier modules. These two nine-digit patterns are stored together to form an eighteen-digit chromosome. The lighter (less intensity) chromosome is assigned to be the cluster centroid of the White Cluster and the other one to be the centroid of the Black Cluster.

Table. I. The pixel $x[i,j]$ with its 8 neighbor pixels

$x[i-1,j-1]$	$x[i-1,j]$	$x[i-1,j+1]$
$x[i,j-1]$	$x[i,j]$	$x[i,j+1]$
$x[i+1,j-1]$	$x[i+1,j]$	$x[i+1,j+1]$

Table. II. Pattern with respect to the pixel $x[i,j]$

$x[i-1,j-1]$	$x[i-1,j]$	$x[i-1,j+1]$	$x[i,j-1]$	$x[i,j]$	$x[i,j+1]$	$x[i+1,j-1]$	$x[i+1,j]$	$x[i+1,j+1]$
--------------	------------	--------------	------------	----------	------------	--------------	------------	--------------

C. Clustering

We have used K-means clustering which was optimized through DE for mapping the known intensities into two clusters Black Cluster and White Cluster. The reason of choosing only two clusters is that in image binarization we are going to map all the pixel intensities either to Black pixels (Intensity: 0) or to White pixels (Intensity: 255).

Each cluster has its own centroid which is a 3×3 matrix (conveying nine dimensional input features). The centroid of each cluster has been chosen randomly from the available $m \times n$ nine-dimensional patterns such that centroids of all the clusters are distinct. Now the clusters are populated with records which are closest to it. The closeness or vicinity is measured between centroid and chosen record through Euclidian distance formula. Finally an output is generated containing information of all the $m \times n$ pixels where all of them have been mapped to black or white intensity.

The clustering takes place using the cluster centroids

generated in the previous step. The distance of pixel from the two centroids using Euclidean distance formula and store the minimum of them in an array along with the cluster they belong to which correspond to the centroid from which the pixel is nearer.

The Euclidean distance between points: p and q is the length of the line segment connecting them. In Cartesian coordinates, if $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ are two points in Euclidean n -space, then the distance from p to q , or from q to p is given by:

$$\begin{aligned}
 d(p, q) &= d(q, p) \\
 &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\
 &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (3)
 \end{aligned}$$

D. Fitness calculation of the chromosome

Each pixel is assigned to a cluster depending upon its distance from the centroids. The pixel is assigned to that cluster whose centroid is closer to the pixel pattern. The aim is to minimize the intra cluster distance for all chromosomes. The intra cluster distance is calculated for every pixel from each cluster using the Euclidean distance formula. The sum of all such distances, i.e., $D_1 + D_2 + D_3 + \dots + D_n$ is the intra cluster distance. The sum of the intra cluster distance of both clusters is the fitness value of the chromosome and we have to minimize this fitness value.

E. Mutation

Every iteration of the differential evolution subjects one of the chromosomes to mutation. This is done to increase the search space. Here apart from the chromosome in consideration three other chromosomes are selected. Among these three the weighted difference of two is added to the third. The weight of the weighted difference is known as the mutation factor, which is a constant between 0 and 2. After doing this operation we get a new chromosome.

F. Recombination

The original chromosome and the newly mutated chromosome are merged together depending on the probability of recombination. The merged chromosome has parts of the original chromosome and parts of newly mutated chromosome. A higher probability of recombination will mean the merged chromosome will have more parts from the mutated chromosome.

G. Selection

The fitness value of the merged chromosome is calculated. If its fitness value is less than that of the original chromosome then the original chromosome is replaced by this new chromosome in the population, else it is discarded. Thus the population size is always preserved.

Steps *E*, *F* and *G* are repeated several times on an initial population to find the optimum chromosome.

In this way the difference frame image is clustered into two classes – one is represented by black cluster for object portion of the image and other is represented by white cluster for background part.

III. EXPERIMENTAL RESULTS

The proposed technique is applied on two video sequences – Hall monitor video and Bowling video. Fig. 1(a) shows the original frames (22nd, 30th, 40th and 55th) of Hall monitor video. Fig. 1(b) displays the result of the corresponding frames by applying Genetic Algorithms [14] (denoted as Method1) and Fig. 1(c) shows the output result of the corresponding frames by using our proposed method.

The original frames (50th, 100th, 130th and 170th) of Bowling video are displayed in Fig. 2(a). Fig. 2(b) displays the result of the corresponding frames by applying Genetic Algorithms [14] (denoted as Method1) and Fig. 2(c) shows the output result of the corresponding frames by using our proposed method.



Fig. 1(a). Original Hall monitor (22nd, 30th, 40th and 55th) frames

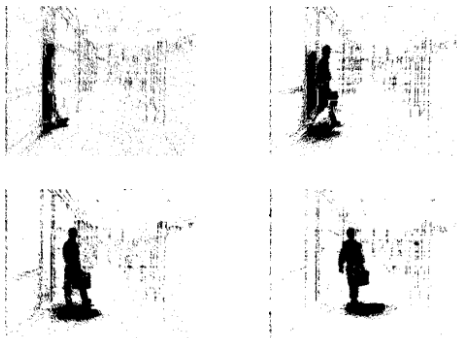


Fig. 1(b). Output results using Method1



Fig. 1(c). Output results using proposed method

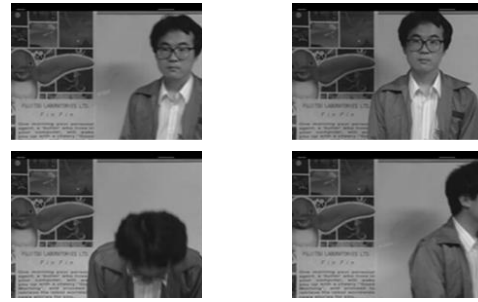


Fig. 2(a). Original frames (50th, 100th, 130th and 170th) of Bowling video



Fig. 2(b). Output results using Method1

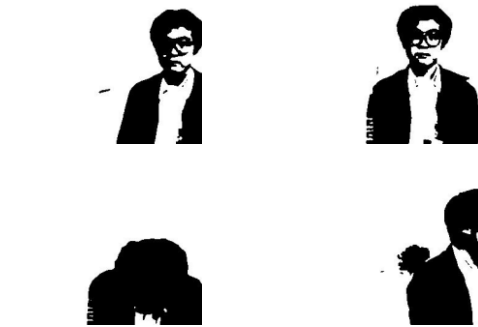


Fig. 2(c). Output results using proposed method

Table. III. Number of misclassified pixels for Hall Monitor video

Frame No.	Method1	Proposed
22	28855	1502
30	20936	1838
40	20092	1558
55	13382	1692

Table. IV. Number of misclassified pixels for Bowling video

Frame No.	Method1	Proposed
50	6795	4296
100	6646	5234
130	1917	1688
170	6363	3743

Moving Object Detection using Differential Evolution

For both the videos the ground truth images are generated manually. Then the results of the proposed algorithm are compared with the ground truth images for both the videos. Table III shows the number of misclassification of the segmented image frames with respect to the corresponding ground truth using Method1 as well as Pro-posed method for Hall Monitor video sequence. Table IV also shows same for the Bowling video. From both the tables, it is found that the proposed method always gives better segmentation result for all the frames than Method1. Since the proposed method gives less number of misclassification compared to the other method.

IV. CONCLUSION

The proposed method successfully detects moving objects in a video with some constraints on the base frame. The camera must remain still so that the base frame does not change. The lighting condition of the base frame should also remain constant. These constraints can be withdrawn if the base frame is continuously updated.

The method has a linear running time and space complexity which is higher than those of conventional methods for detecting moving objects (using normal frame differencing) due to the presence of much involved computations in finding the cluster centres and optimizing them using Differential Evolution Algorithm. However the method is expected to produce better results even in the presence of pixel noise. Pixel noise is mostly due to the use of low resolution cameras or insufficient illumination. Thus it is expected that the use of low resolution cameras would compromise with the high time and space complexity to produce results with tolerable complexities.

REFERENCES

1. K. Skifstad, R. Jain and C. Ramesh, "Illumination independent change detection for real world image sequences", *Computer Vision, Graphics and Image Processing* 46(3), pp. 387-399, (1989).
2. Y. Z. Hsu, H. H. Nagel and G. Rekers, "New likelihood test methods for change detection in image sequences", *Computer Vision, Graphics and Image Processing*, 26(1), pp. 73-106, (1984).
3. T. Aach, A. Kaup and R. Mester, "Statistical model-based change detection in moving video", *Signal Processing* 31, pp. 165-180, (1993).
4. K. McKoen, R. Navarro-Prieto, B. Duc, E. Durucan, F. Ziliani and T. Ebrahimi, "Evaluation of video segmentation methods for surveillance applications", *EUSIPCO 2000*, Tampere, Finland, (Sept. 2000).
5. P. Villegas, X. Marichal and A. Salcedo, "Objective evaluation of segmentation masks in video sequences", *WIAMIS'99 Workshop*, Berlin, Germany, (May 1999).
6. M. Wollborn and R. Mech, "Procedure for objective evaluation of VOP generation algorithms", *Fribourg, ISO/IEC JTC1/SC29/WG MPEG97/2704*, (1997).
7. E. Durucan and T. Ebrahimi, "Change Detection and Background Extraction by Linear Algebra", *Proceedings of the IEEE* 89(10), (October 2001).
8. K. V. Price, "An Introduction to Differential Evolution", *New Ideas in Optimization*, McGraw-Hill, London, pp. 79-108, (1999).
9. R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *Journal of Global Optimization* 11, pp. 341-359, (1997).
10. S. Z. Li, "Markov Random Field Modeling in Image Analysis", *New York: Springer*, (2001).
11. E. Y. Kim and K. Jung, "Genetic Algorithms for video Segmentation", *Pattern Recognition* 38(1), pp. 59-73, (2005).
12. E. Y. Kim and S. H. Park, "Automatic video Segmentation using genetic algorithms", *Pattern Recognition Letters* 27(11), pp. 1252-1265, (Aug. 2006).
13. S. W. Hwang, E. Y. Kim, S. H. Park and H.J. Kim, "Object Extraction and Tracking using Genetic Algorithms", *Proceedings of International Conference on Image Processing* 2, pp. 383-386, (2001).

14. A. Raychaudhuri and M. De, "A Novel Approach of Detection of Moving Objects in a Video", *International Journal of Advanced Research in Computer and Communication Engineering* 2(11), pp. 4485 - 4488, (2013).

AUTHORS PROFILE



Amlan Raychaudhuri received the B.Sc. degree (with honours) in Computer Science from the University of Calcutta, Kolkata, India, in 1999. He also received the M.Sc. degree in Computer and Information Science and the M.Tech. degree in Computer Science and Engineering from the University of Calcutta, Kolkata, India, in 2001 and 2003, respectively.

He was a Lecturer with the Department of Computer Science and Engineering, B. P. Poddar Institute of Management and Technology under West Bengal University of Technology, Kolkata, India, during 2004 - 2008. He is currently an Assistant Professor at the same department. His research interest include soft computing, image processing, video processing and pattern recognition.



Arkadev Roy is a final year B.Tech student in Computer Science & Engineering at B. P. Poddar Institute of Management and Technology, under West Bengal University of Technology, Kolkata. His interests include Cloud computing, web technology, image processing amongst others.



Ashesh Das is a final year B.Tech student in Computer Science and Engineering at B. P. Poddar Institute of Management and Technology, under West Bengal University of Technology, Kolkata. Apart from computer science he is also interested in fields like Image Processing, Robotics and Video Editing.



Gourav Kumar Shaw is a final year B.Tech student in Computer Science and Engineering at B. P. Poddar Institute of Management and Technology, under West Bengal University of Technology, Kolkata. His interests are Image Processing, Embedded Systems and Robotics.



Pratik Kumar Mitra is pursuing his B.Tech from B. P. Poddar Institute of Management and Technology, under West Bengal University of Technology, Kolkata and is in his final year. His areas of interest include Image Processing, Computer Graphics and Robotics.