

A New Algorithm for Tracking Objects in Videos of Cluttered Scenes

Andres Alarcon Ramirez¹ and Mohamed Chouikha²

¹Department of Computer Engineering, Howard University, Washington, DC, USA

alarconramirezandr@bison.howard.edu

²Department of Computer Engineering, Howard University, Washington, DC, USA

mchouikha@howard.edu

ABSTRACT

The work presented in this paper describes a novel algorithm for automatic video object tracking based on a process of subtraction of successive frames, where the prediction of the direction of movement of the object being tracked is carried out by analyzing the changing areas generated as result of the object's motion, specifically in regions of interest defined inside the object being tracked in both the current and the next frame. Simultaneously, it is initiated a minimization process which seeks to determine the location of the object being tracked in the next frame using a function which measures the grade of dissimilarity between the region of interest defined inside the object being tracked in the current frame and a moving region in a next frame. This moving region is displaced in the direction of the object's motion predicted on the process of subtraction of successive frames. Finally, the location of the moving region of interest in the next frame that minimizes the proposed function of dissimilarity corresponds to the predicted location of the object being tracked in the next frame. On the other hand, it is also designed a testing platform which is used to create virtual scenarios that allow us to assess the performance of the proposed algorithm. These virtual scenarios are exposed to heavily cluttered conditions where areas which surround the object being tracked present a high variability. The results obtained with the proposed algorithm show that the tracking process was successfully carried out in a set of virtual scenarios under different challenging conditions.

KEYWORDS

Video object tracking, region of interest, cluttered conditions

1. INTRODUCTION

Video object tracking can be defined as the detection of an object in the image plane as it moves around the scene. This topic has a growing interest for both civilian and military applications, such as automated surveillance, video indexing, human-computer interaction (gesture recognition), meteorology, and traffic management system [1][2][3].

There are two basic problems that a tracking system must resolve: the motion estimation and the matching estimation. The motion estimation predicts the location of the most likely region in the next video frame where the object being tracked may be placed. Commonly, this information is not available; therefore, a mechanism to determine the fixed-size region surrounding the object being tracked is needed. A technique widely used to resolve this problem in video tracking is Kalman Filter (KF), which is an optimal recursive estimator of the state of a dynamic system. On the other hand, matching estimation seeks to identify an object, which is being tracked in the current video frame, inside a closed region in the next video frame. The closed region is predicted

by the motion estimation stage and corresponds to the zone where there is the highest probability of finding the object of interest in the next video frame. The exact location of the object of interest in the next frame is determined in the matching estimation stage by using information extracted from the object itself in previous scenes. Matching estimation algorithms incorporate a feature detection stage that constitutes the first step in different processing operations such as image classification and segmentation. Feature detection is used by object tracking algorithms to carry out a matching of the pixels from the object being tracked between two consecutive video frames, and then determine the exact location of the object in the next frame.

The algorithm presented in this paper is an automatic video object tracking algorithm that uses a region of interest (ROI) defined completely inside the object being tracked to carry out the tracking process. To do this, the proposed algorithm guarantees that when the object to be tracked suffers displacements, the ROI rebounds against the inner walls of the object being tracked and stays inside this object. On the other hand, the paper also presents the designing of a software platform that is used to create virtual scenarios where objects with different shapes and sizes wander into scenes through the time. These virtual scenarios are useful to test the performance of the proposed object tracking algorithm under different challenging problems such as cluttered conditions and random movement.

This paper is organized as follows: In the Section 2, we describe the related work, in the Section 3, we present the novel proposed algorithm to track an object through a video. The section 4, on the other hand, describes the software platform used to create virtual scenarios to test video object tracking algorithms. In the Section 5, it is shown the obtained results. Finally, the Section 6 presents the conclusions of this work.

2. RELATED WORK

The process of tracking an object in a sequence of frames is directly dependent on the object's representation being used. Some representations, for example, use interest points to identify the object to be tracked [4]. These interest points can be detected by using information based on differentiation operators [5][6], where changes in intensity between two adjacent pixels can emphasize the boundaries of the object of interest in the image [6]. Other object's representations use its silhouette or contour to extract information about the general shape of the object [7] [8]. Once the contour or the inner region of a given object is identified, different characteristics are then extracted and used as features to be tracked between two consecutive frames.

Cross-correlation, on the other hand, was used in [9] to implement a face tracking algorithm for video conferencing environment. This method compares a region of the image with a known signal extracted from the object of interest, and then a measure of similarity, which allows to determine the exact position of the object being tracked in the next frame, is obtained between the two signals. In [10] is presented a methodology for video object tracking that is constituted by four steps, namely, background subtraction, candidate object identification, target object selection, and motion interpolation. Firstly, the background, which is available, is subtracted to the current frame to identify the object being tracked; additionally, the background is updated from time to time whenever there is a permanent change in it. Then, a threshold is applied over the image, which is the result of subtracting the current frame from the background, to generate a new binary image with the candidate objects from which the target object is selected using histogram matching. Finally, motion interpolation determines the displacement of the object from one frame to another.

Mean-shift (MS) is another technique of video object tracking that is based on primitive geometric shapes [11]. At the beginning, it is defined a region of interest around the object to be

tracked in the current frame, and then it is started an iterative process based on comparing the histogram of the region of interest in the current frame with the histograms obtained from candidate regions in the next frame where there exist the chances of finding the object being tracked. Finally, the location of the object being tracked in the next frame is defined by the candidate region whose histogram presents the greatest similarity with the histogram from the region of interest which surrounds the object in the current frame.

Scale-Invariant Feature Transform (SIFT), which is used for finding local points [12], was integrated to MS to create a methodology that jointly employs point feature correspondences and object appearance similarity [13]. Chakraborty and Patra presented a Kernel-based algorithm that uses segmentation techniques to determine the target localization [14]. Babu *et al.* [15], instead of using a single mean shift tracker, used multiple mean shift tracking points.

Chun-Te *et al.* [16] used projected gradient to help multiple inter-related kernels in finding the best match during tracking under predefined constraints. On the other hand, multiple kernels were incorporated into a Kalman filtering-based tracking system [17]. In their design, not only the state transition matrix but also the noise covariance matrix used in Kalman filtering is dynamically updated. Liu *et al.* [18] proposed a pixel classification approach based on Markov random field MRF to track objects in video sequences, where kernel density estimation founded on nonparametric models was used to represent both video objects and background. Additionally, spatial context and temporal coherency modeled in MRF are exploited to ensure a more robust segmentation performance.

Hossein and Bajie proposed a framework [19] for tracking moving objects based on spatio-temporal Markov Random field, and where are taken into account the spatial and temporal aspects of the object's motion. Amer presented an automatic object tracking algorithm based on the matching of features in successive frames [20]. Initially, the objects being tracked are segmented and their spatial and temporal features are computed. Then, using a nonlinear two-stage voting strategy, each object of the previous frame is matched with an object of the current frame creating a unique correspondence.

However, the previous tracking algorithms have shown to have difficulties when tracking objects in videos of cluttered scenes. Namely, when the pixels from the region around the object being tracked present intensity variations across the video.

3. TRACKING APPROACH

The proposed algorithm is constituted by two important stages: the stage of estimation of the direction of the object's motion and the matching process. The first stage analyzes the changing areas generated as result of the object's motion, specifically in regions of interest defined inside the object being tracked in both the current and the next frame. The latter stage determines the location of the object being tracked in the next frame by evaluating a function of dissimilarity that is minimized using information extracted from the object being tracked in the current video frame.

3.1. Estimation of the Direction of the Object's Motion

The proposed new strategy seeks to simplify the problem of tracking. To do this, the direction of the object's motion is estimated by employing the changing areas generated from the object's motion itself between the current frame and the second frame, and where only the pixels belonging to two regions of interest defined in the first and the second frame respectively are taken into account. The region of interest in the first frame is defined in such a way that this

region is completely inside the object of interest, whereas the second region of interest is defined in the same coordinates than the first one but in the second frame.

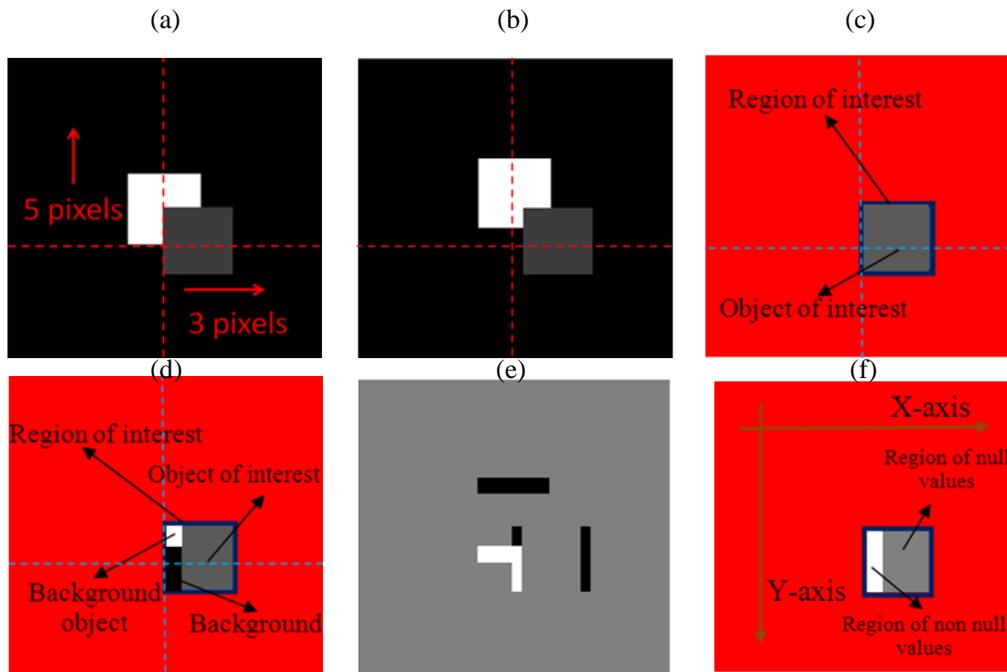


Figure 1: Motion of a background object and the object to be tracked between two consecutive images, (a) Current frame of the video, (b) next frame of the video, (c) region of interest located in the first frame, (d) region of interest located in the second frame, (e) resulting image of subtracting two consecutive images, (f) region of interest located in the resulting image of subtracting two consecutive image.

The Fig. 1 (a) and the Fig 1 (b) show a simple scenario where both a background object and the object being tracked are displaced between the first and the second frame. Additionally, it is shown the location of the region of interest in the first frame and the second frame respectively. The Fig. 1 (c) shows that the environment around the object of interest in the first frame is ignored. Similarly, the Fig. 1 (d) shows the zone in the second frame where the object of interest, the background object, and the background are delimited by the region of interest.

The Fig. 1 (e), on the other hand, shows the resulting image of subtracting the first frame shown in the Fig. 1 (a) of the second frame shown in the Fig. 1 (b). This resulting image presents two regions of pixels of non-null values on a background of pixels of null values; the regions with non-null pixels are generated as result of the movement of both the object of interest and the background object. Nevertheless, these regions of pixels as a whole offer no clue about the nature of the movement carried out by the object of interest.

On the other hand, if we define the same region of interest used in the first two frames but in the image shown in the Fig. 1 (e), we will obtain a new image which reduces considerably the complexity of the image shown in the Fig. 1 (e). This new image, which is shown in the Fig. 1 (f), only takes into account the behaviour of the object being tracked in the region of interest, and ignores the environment that surrounds the region of interest. To determine the direction of the object's motion, we select the region of interest defined in the object being tracked, and then it is constituted two sets of pixels such as follows:

$$S_1 = \{(x, y) | (x, y) \in R \text{ and } F(x, y) \neq 0\} \tag{1}$$

Thus, the set, S_1 , represents the group of coordinates of the pixels in the region of interest, R , where the image, $F(x,y)$, has non-null values, i.e., $F(x,y) \neq 0$. Note that, $F(.)$, represents the resulting image of subtracting two consecutive images. The second set is constituted by the coordinates of all pixels which are in the region of interest such as follows:

$$S_2 = \{(x, y) | (x, y) \in R\} \tag{2}$$

The groups of pixels which constitute the sets, S_1 and S_2 , are shown in the Fig. 2 (a) and the Fig. 2 (b) respectively.

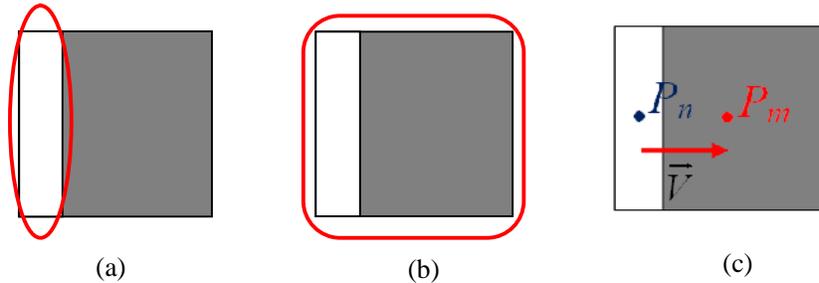


Figure 2: Analysis of the region of interest in the resulting image of subtracting two consecutive frames, (a) region of pixels used to calculate the centroid P_n , (b) region of pixels used to calculate the centroid P_m , (c) locations of the centroids P_n and P_m ; vector that defines the direction of object's motion.

On the other hand, if we use the Equation (3) to calculate the average of the coordinates that constitute the group, S_1 , which was defined by the Equation (1), we will obtain the point, P_n , which represents the centroid of the group, S_1 .

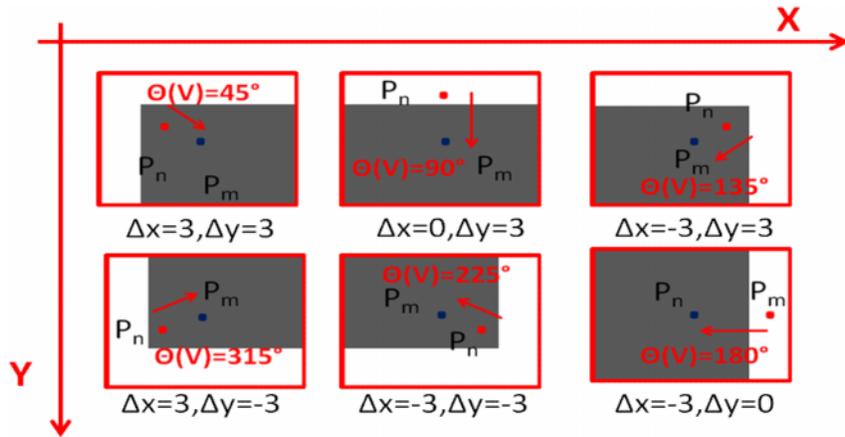


Figure 3: Vectors obtained under different directions of motion of the object being tracked.

In the same way, the centroid of the group, S_2 , which is represented by the point, P_m , is calculated using the Equation (4). The locations of the centroids, P_n and P_m , in the region of interest are shown in the Fig. 2 (c).

$$P_n = (\overline{x_n}, \overline{y_n}) = \frac{1}{N} * \sum_{(x_i, y_i) \in S_1}^N (x_i, y_i) \quad (3)$$

$$P_m = (\overline{x_m}, \overline{y_m}) = \frac{1}{N} * \sum_{(x_i, y_i) \in S_2}^N (x_i, y_i) \quad (4)$$

The two points, which correspond to the centroids of the sets, S_1 and S_2 , constitute a vector whose direction determines the orientation of the object's motion. In other words, the vector which connects the centroid, P_n , to the centroid, P_m , has the same direction than the object being tracked. This vector is defined by the following equation.

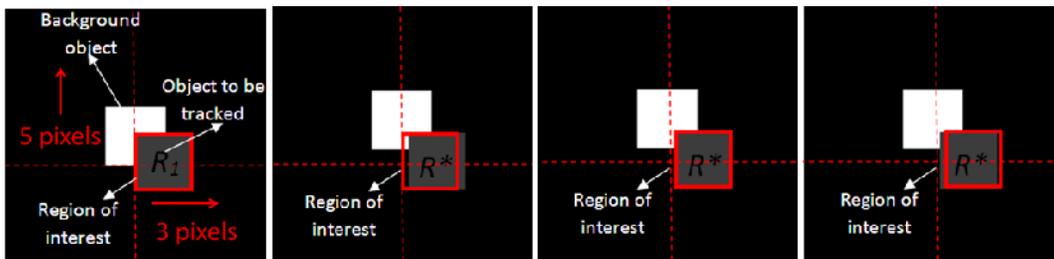
$$\vec{V} = P_m - P_n \quad (5)$$

Finally, the angle of the vector which determines the direction of the object's motion is calculated using the Equation (6).

$$\theta = angle(\vec{V})_I \quad (6)$$

The Fig. 3 presents the regions of interest in the images generated from subtracting two consecutive frames, and where the object being tracked is moved in different orientations. Each of these regions shows the location of the centroids, P_n and P_m , which were calculated from the sets, S_1 and S_2 respectively. In the same way, the two centroids constitute the vector which determines the orientation of movement of the object being tracked.

Once the direction of the object being tracked has been determined, it is started an iterative process which seeks to determine the location of the object being tracked in the next frame (second frame). To do this, it is used the region of interest which was defined in the current frame (first frame), R_I , to determine the direction of the object's motion. This region of interest is located totally inside the object being track such as it is shown in the Fig. 4 (a). At the same time, it is defined in the second frame a second region of interest, R_* , with the same shape, size, and location of the first region of interest used in the current frame. The initial location of the region of interest, R_* , in the second frame is shown in the Fig. 4 (b).



During each iteration of the iterative process, the region of interest, R_* , is displaced across the second frame in the direction of the object's motion which was initially estimated. The Equations (7) and the Equation (8) describe the movement of the region of interest, R_* .

$$\Delta x = k * \cos(\theta) \quad (7)$$

$$\Delta y = k * \sin(\theta) \quad (8)$$

where, k is an integer which takes the values of $0,1,2,\dots, S$. The parameter, S , is a constant which represents the maximum possible displacement of the object being tracked; its value is defined by the user and depends on the nature of the video being processed. Finally, the parameter, θ , is the angle of the vector which defines the direction of the object's motion. This angle is calculated using the Equation (6).

On the other hand, at each iteration of the iterative process which seeks to establish the displacement of the object being tracked, the region of interest, R_* , located in the second frame is compared with the region of interest, R_I , located in the first frame. The Equation (9) presents the function of dissimilarity, $M(x, y)$, which is used to compare these two regions.

$$M(\Delta x, \Delta y) = \sum_{i=1}^L \left[U \left(\begin{matrix} R_1(x_i, y_i) - \\ R^*(x_i + \Delta x, y_i + \Delta y) \end{matrix} \right) \right] \quad (9)$$

where the parameter, L , represents the number of pixels which constitutes the region of interest R_I . The function, $U(\cdot)$, is defined by the Equation (10).

$$U(x) = \begin{cases} x = 0 & 0 \\ x \neq 0 & 1 \end{cases} \quad (10)$$

It is important to mention that that the function, $M(\cdot)$, depends on the parameters, x and y , which correspond to the horizontal and vertical displacements of the second region of interest, R_* . In general, the function, $M(\cdot)$, compares one-to-one the pixels located in the first region, R_I , with the pixels located in the second region, R_* , to determine the number of different pixels between these two regions. Thus, to the end of the iteration process, we will obtain a set of values for the function, $M(\cdot)$, for different values of displacement. The Fig. 4 (b) to (f) present the graphical representation of the movement of the region of interest, R_* , across the second frame following the direction of the object's motion.

To the end of the iterative process, it is selected the pair of values for x and y that after being evaluated in the function, $M(\cdot)$, obtains the minimum value for this function among a total of ' S ' possible values. The pair of values selected defines the displacement of the region of interest, R_* , which corresponds to the same displacement carried out by the object being tracked between the current and next frame. The iterative procedure described above constitutes a process of minimization that can be described by the following expression:

$$\text{Min} \left[M(\Delta x, \Delta y) = \sum_{i=1}^L \left[U \left(\begin{matrix} R_1(x_i, y_i) - \\ R^*(x_i + \Delta x, y_i + \Delta y) \end{matrix} \right) \right] \right] \quad (11)$$

At the end of the process of minimization, the region of interest in the current frame is updated to the new location defined by the pair of values, x and y , that minimizes the function, $M(\cdot)$, such as follows:

$$Y_i = Y_i + \Delta y \quad (12)$$

$$X_i = X_i + \Delta x \quad (13)$$

where the pair of coordinates, X_i and Y_i , corresponds to the location of the center of the region of interest in the current frame. On the other hand, the second frame becomes in the current frame, and the new frame in the sequence of images (third frame) becomes in the next frame. Thus, the updated location of the region of interest defines the position of the object being tracked.

Finally, the proposed algorithm can be summarized in the following steps:

```

Determine  $\theta$ , (Equations 1-6)
Define the value of S
For k=0 to S do

     $\Delta x_k = k * \cos(\theta)$ 
     $\Delta y_k = k * \sin(\theta)$ 
     $M(\Delta x, \Delta y) = \sum_{i=1}^L \left[ U \left( R_1(x_i, y_i) - R^*(x_i + \Delta x, y_i + \Delta y) \right) \right]$ 

end

Select  $\left\{ \begin{array}{l} \Delta x_k, \Delta y_k \mid \text{Min } M_k(\Delta x_k, \Delta y_k), \\ \forall k \in \{0, 1, 2, \dots, S\} \end{array} \right\}$ 

Update the location of  $R_j$ 
    
```

4. DESIGN OF A SOFTWARE PLATFORM TO CREATE VIDEOS OF TESTING

To test the performance of the proposed algorithm, it is necessary to create a set of virtual scenarios where objects with different shapes and sizes wander into scenes through the time. Therefore, it is designed a software platform that is able to create a wide range of videos where not only the object to be tracked is present in the scenes but also background objects. These background objects may be motionless or they may be moved in different directions through the video. Additionally, the software platform is able to create video sequences under heavily cluttered conditions, meaning by cluttered conditions the presence of a considerable number of background objects that interact with the object being tracked. The cluttered conditions constitute an important challenge for tracking algorithms because the region that surrounds the object of interest presents a high variability that hinders the modeling of the tracking process in general. The software platform, which was designed in Matlab 7.12.0, is able to create a wide variety of scenarios by setting the following parameters:

Number of objects presents in the video. The number of objects which will be located in the scenes may be defined by the user or may be selected automatically by the software platform.

Shapes and sizes of the objects present in the video. The user may select among a diverse set of geometric figures the shapes of each object present in the video. Among these geometrical forms, we have rectangles, triangles, ovals, and other forms composed of two or more of these geometrical forms.

Levels of intensity of each object present in the video. In the same way, the gray-level intensity of each object located in the video is selected by the user or defined randomly by the software platform as a value which ranges from 0 to 255.

Location of each object in the video. The coordinates of each background object in the first frame of the video may be initialized by the user or defined randomly by the testing platform.

Movements adopted by the objects in the video. The background objects may be motionless through the video or they may be moved around along with the object of interest according to one out of three strategies, i.e., a random movement, where it is selected randomly two numbers whose values ranges from 0 to 5 pixels. The two values represent the x-axis and y-axis displacements of the object in the next frame. The uniform distribution, which is defined by the Equation (20), is used to generate the displacement values. Once the object has been displaced, two new random values are obtained to move the object in the next frame. This process is repeated for the total number of images which constitute the video.

$$f(x) = \begin{cases} \frac{1}{J} & 0 \leq x \leq J \\ 0 & \text{Otherwise} \end{cases} \quad (14)$$

The parameter, J , in the Equation 14, represents the maximum displacement that the object may carry out. The second strategy corresponds to a predefined displacement, and it consists in selecting two random values for the x-and y-axis displacements of the object being tracked in the same manner as was done for the first strategy of random movement described above. However, these two values are adopted as fixed displacement of the object of interest between two consecutive frames, and for all the sequence of images which constitute the video. In other words, the displacement of the object is always the same between two consecutive images. The last strategy is a combination of the previous two, where the displacement of the object has a 50 percent chance of being completely random (first strategy) and 50 percent chance of being a predefined displacement (second strategy).

On the other hand, the frames generated have a size of 600 x 600 pixels, and every video is constituted by 100 frames. Different geometrical shapes as squares, rectangles, ovals, triangles, and some combinations of the previous ones were used for the object to be tracked.

5. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in Matlab 7.12.0, and different tests were carried out in a PC, Dell Inspiron 640 m, with 2 GB of RAM memory. Initially, it is defined a region of interest, R , in the first frame of the video. This region is placed in such a way that it is completely inside of the object to be tracked. Once the location of the region of interest has been defined in the first frame of the video sequences, the proposed algorithm updates automatically the location of this region of interest for all the remaining video frames. In other words, the location of the object of interest is automatically identified through the video by the proposed algorithm. Additionally, the size of the regions of interest used for all the experiments is of 20 x 20 pixels. Finally, the results obtained by the proposed algorithm with the set of virtual videos are shown form the Fig. 5 to the Fig. 8.

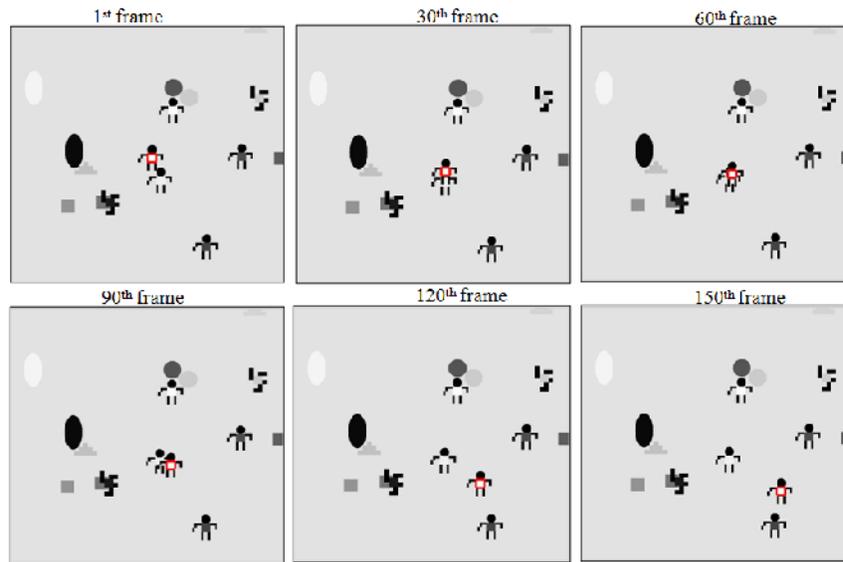


Figure 5: Tracking of an object in the first video sequence

The Fig. 5 shows six particular frames from a video sequence constituted by 150 frames. The video sequence shown in the Fig. 5 presents a group of object moving in different directions. The object to be tracked is selected in the first frame by placing a red square-shaped region in the object. Then, the algorithm automatically updates the position of this square region in such a way that this region move along with the object through the video sequence allowing the tracking of the object of interest.

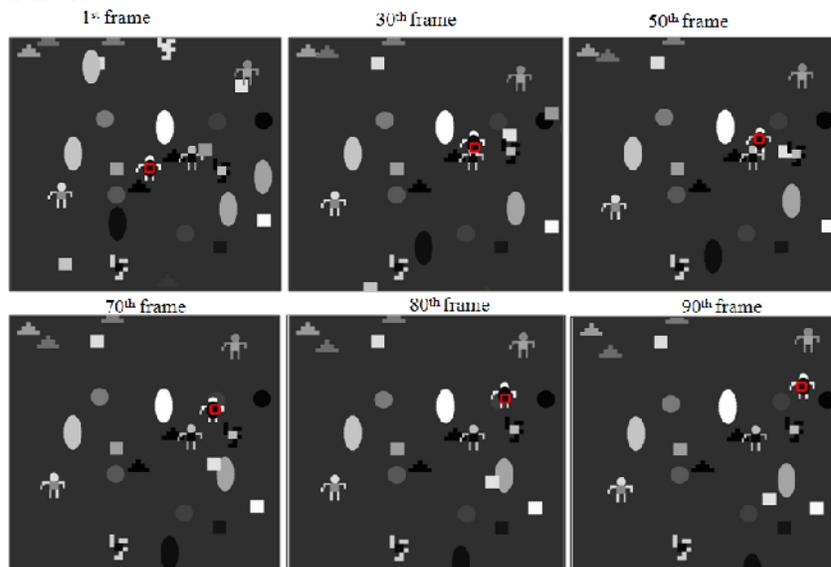


Figure 6: Tracking of an object in the second video sequence

Similarly, the Fig. 6 shows the tracking of an object in a video sequence constitutes by 90 frames. Once again, the object to be tracked is selected in the first frame, and then the proposed algorithm automatically locates it in the rest of the video sequence.

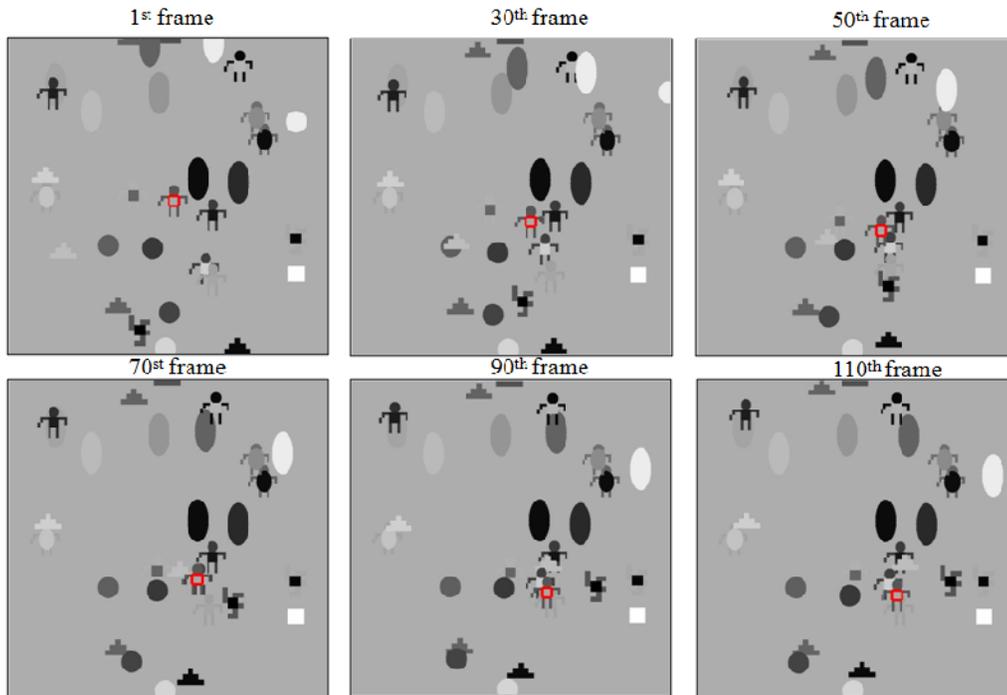


Figure 7: Tracking of an object in the third video sequence

The Fig. 7 shows another video sequence constituted by a group of objects moving randomly in time. The object to be tracked is selected and identified by a red square-shaped region that is placed in the object. The results obtained in this video sequence show that the object of interest is successfully tracked.

Finally, the Fig. 8 shows the results obtained by applying the proposed algorithm to a video sequence of 100 images. Each image has approximately 100 background objects that interact with the object being tracked in different ways. Once again, the object being tracked, which is identified by a red square-shaped region, is successfully located in the video.

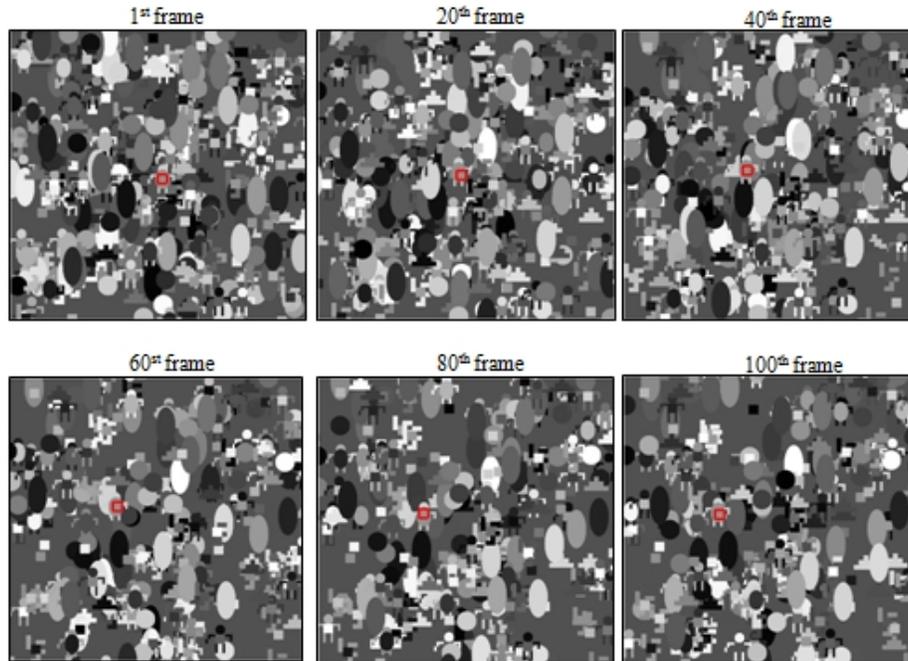


Figure 8: Tracking of an object in the fourth video sequence

The obtained results shows that the proposed algorithm can track several objects in scenarios with heavily cluttered conditions, where the region which surrounds the object of interest experiences continuous changes because of the interaction between the object being tracked and the background objects located in the scenes.

6. CONCLUSIONS

The novel algorithm proposed in this paper to track an object in video sequence was successfully tested under a wide variety of scenarios where heavily cluttered conditions were present. Additionally, it was designed a testing platform which allowed to create challenging scenarios used to test the proposed tracking algorithm. This testing platform was an important tool at the beginning of the design of the proposed algorithm as well as in the analysis of the tracking process in general, and it becomes in an important instrument to study in future works other phenomena presented in video object tracking such as occlusion, scale changes, illumination changes, etc. On the other hand, the proposed algorithm based on region of interest stayed immune to heavily cluttered conditions because this algorithm ignores much of the variability in the environment which surrounds the object being tracked.

REFERENCES

- [1] G. L. Foresti, Object Recognition And Tracking For Remote Video Surveillance, IEEE Trans. Circuits Syst. Video Technol., 9(7):1045-1062, October 1999.
- [2] A. J. Lipton, H. Fujiiyoshi, R. S. Patil, Moving Target Classification And Tracking From Real-time Video, Applications of Computer Vision,1998. WACV '98. Proceedings., Fourth IEEE Workshop on, pp. 8-14, 1998.
- [3] Y. Li, A. Goshtasby, and O. Garcia, Detecting and tracking human faces in videos, Proc. ICPR'00 vol. 1, pg. 807-810 (2000).
- [4] Gabriel, P.; Hayet, J.-B.; Piater, J.; Verly, J., "Object Tracking using Color Interest Points," IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.

- [5] C.Harris; M.Stephens., "A combined corner and edge detector," 4th Alvey Conference, pages 147–151, 1988.
- [6] J.J. Koenderink and A.J. Van Doorn,"Representation of local geometry in the visual system," *Biological Cybernetics*,55(6), 1987.
- [7] Haritaoglu, I., Harwood, D., and Davis, L., "real-time surveillance of people and their activities," *IEEE Trans. Patt. Analy. Mach. Intell.* 22, 8, 2000.
- [8] Sato, K. and Aggarwal, J., "Temporal spatio-velocity transform and its application to tracking and interaction," *Comput. Vision Image Understand.* 96, 2, 100–128,2004.
- [9] Sebastian, P.;Yap Vool Voon,"Tracking using normalized cross correlation and color space,"*International Conference on Intelligence and Advanced system, ICIAS*, 2007.
- [10] Ashwani Aggarwal, Susmit Biswas, Sandeep Singh, Shamik Sural, and A.K. Majumdar, " Object Tracking. Using Background Subtraction and Motion Estimation in MPEG Videos", *ACCV. 2006, LNCS*, vol. 3852, pp. 121-130, Springer, Heidelberg (2006).
- [11] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Proc. on Computer Vision and Pattern Recognition*, pages 673–678, 2000.
- [12] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int'l Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [13] Khan, Z.H., Gu, I.Y.-H., TieSheng Wang, Backhouse, A., "Joint anisotropic mean shift and consensus point feature correspondences for object tracking in video,"*IEEE International Conference on Multimedia and Expo*, 2009.
- [14] Chakraborty, D.; Patra, D.,"Real time object tracking based on segmentation and Kernel based method," *IEEE International Conference on Industrial and Information systems*, 2010.
- [15] R. V. Babu, P. Perez, and P. Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," *Image and Vision Computing*, vol. 25, issue 8, pp. 1205-1216, 2007.
- [16] Chun-Te Chu; Jenq-Neng Hwang; Hung-I Pai; Kung-Ming Lan, "Robust video object tracking based on multiple kernels with projected gradients," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.
- [17] Chun-Te Chu, Jenq-Neng Hwang, Shen-Zheng Wang, Yi-Yuan Chen, "Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients," *IEEE International Conference on Distributed smart Cameras*, 2011.
- [18] Zhi Liu; Liquan Shen; Zhongmin Han; Zhaoyang Zhang, "A Novel Video Object Tracking Approach Based on Kernel Density Estimation and Markov Random Field ," *IEEE International Conference on image processing*, 2007.
- [19] Khatonabadi, S.H.; Bajic, I.V., "Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields," *IEEE Transaction on Image Processing*, 2013.
- [20] Amer, A., "Voting-based simultaneous tracking of multiple video objects," *IEEE Transaction on Circuits and Systems for Video Technology*, 2005.

AUTHORS

Andres Alarcon-Ramirez received a PhD degree in Electrical and Computer Engineering from Howard University. He received his M.S. in Computer Engineering from University of Puerto Rico (2009), where he was a research assistant for The Bernard M. Gordon Center for Subsurface Sensing and Imaging Systems. He also received a M.S. in Electrical Engineering (2006), and his BS in Electrical Engineering (2003) from Universidad del Valle (Cali-Colombia). Currently, he is working as research assistant at the Electrical Department at Howard University.



Mohamed F. Chouikha (M '88) received a Ph.D. degree in Electrical Engineering from the University of Colorado in Boulder in 1988. Since 1988, he has been with Department of Electrical Engineering at Howard University. In July 2000, he became the Chair of the EE Department and had since held the position. Dr. Chouikha's research interests include multimedia signal processing and communications, and wireless Communications.

