



Feature Extraction Algorithm for Real Time Object Tracking

{A.M. Sallam^{*}, O.M. Elmowafy, R.A. Elbordany and A.M. Fahmy}[†]

Abstract: Video tracking system raises a wide possibility in today's society. These systems are used in various applications such as military, security, monitoring, robotic, and nowadays in day-to-day applications. However the video tracking systems still have many open problems and various research activities in the video tracking system are explored. This paper presents a real time algorithm for video tracking of any moving target with the use of line profile moments within a window filter. The proposed system is suitable for indoor and outdoor applications. Our approach has the advantage of extending the applicability of tracking system and also, as presented here improves the performance of the tracker making feasible high frame rate video tracking. The goal of the tracking system is to analyze the video frames and estimate the position of a part of the input video frame (usually a moving object), our approach can detect and track any moving object and calculate its position. Therefore, the aim of this paper is to construct a motion tracking system for the moving object. Where, at the end of this paper, the detail outcome and results are discussed using experimental results of the proposed technique.

Keywords: Real-time image processing, tracking system, image tracking, line profile moments, video tracking, window filter tracking.

1. Introduction

The problem of object tracking can be considered an interesting branch in the scientific community and it is still an open and active field of research [1], [2]. This is a very useful skill that can be used in many fields including visual serving, surveillance, gesture based human machine interfaces, video editing, compression, augmented reality, visual effects, motion capture, medical and meteorological imaging, etc... [3], [4].

In the most approaches, an initial representation of the to-be-tracked object or its background is given to the tracker that can measure and predict the motion of the moving object representation overtime.

The major restriction for this paper is to detect and track object in real time, we need a fast and simple algorithm for segmentation to achieve tracking in real time, for that reason we need to use the thresholding technique. In general, most of the tracking algorithms solve the tracking problem in off-line trajectory application or in run-time application [5]. In this paper the proposed algorithm try to solve the tracking problem in the real-time using feature-based video object tracking [6], [7] with (Visual C++).

^{*} Egyptian Armed Forces, ahmed_wesam@hotmail.com

[†] Egyptian Armed Forces, Egypt.

Real-Time object tracking is a very specific field of study within the general scope of image processing and analysis. Human can recognize and track any object perfectly, instantaneously, and effortlessly even the presence of high clutter, occlusion, and non-linear variations in background, target shape, orientation and size. However, it can be an overwhelming task for a machine! There are partial solutions, but the work is still progressing toward a complete solution for this complex problem [8].

In the remaining part of this paper, we will explain our literature review in Section 2. Then, in Section 3, we describe the desirable system features and algorithms necessary for successful system. In section 4 we describe the system architecture (implementation environment of the system), and the proposed algorithm that will be used in our method. In Section 5, the experimental results and comparison between the proposed algorithm and the temporal filtration algorithm are presented. Finally, in Section 6 we will discuss and analyze the results obtained from section 5.

2. Tracking System: a Literature Review

In the recent times the vast number of algorithms has been proposed in the field of object tracking. An even greater number of solutions have been constructed from these algorithms, many solving parts of the puzzle that makes computer vision so complex.

One technique proposed is to use the small chromatic-space of human skin along with facial such as eyes, mouth and shape to locate faces in complex color images. Yang and Ahuja [9] investigated such an object localization techniques, where experimental results concluded, “human faces in color image can be detected regardless of size, orientation or viewpoint.” It was illustrated that the major difference in skin color across different appearances was due to intensity rather than color itself.

McKinnon [10] also used in similar skin filtration based theory to implement a multiple object tracking system. McKinnon stated that his solution was often limited by the quality of the skin sample supplied initially. Further to this, in real-time environment the lack of or excessive level of light could cause the performance to suffer.

The drawback of skin color systems is that they can only track objects containing areas of skin-color-like; areas in the background may be confused with real regions of interest. As such they are not suitable for use in all applications and hence are often limited in their use [11].

The most two popular methods for image segmentation used in the object tracking field are temporal segmentation and background subtraction. Vass, Palaniappan and Zhuang presented a paper [12] that outlined a method of image segmentation based on a combination of temporal and spatial segmentation. By using interframe differences [13] a binary image was obtained showing pixels that had undergone change between frames. Temporal segmentation on its own fails for moving homogeneous regions, as such spatial segmentation was incorporated. Using a split and merge techniques an image are split into homogenous regions. Finally, by merging spatial and temporal information, segmentation of motion areas was achieved at a rate of approximately five frames per second; however a small amount of background was evident in the resulting segmented regions.

Background subtraction could be considered similar to temporal filtration. The difference being, background subtraction computes the difference between an empty reference frame and future frames while temporal filtration computes the difference between consecutive frames.

Andrews [14] utilized background subtraction to create a system based on distance measures between object shapes for real-time object tracking. By acquiring an initial image of the operational environment free of moving objects, he was able to cleanly segment areas of change in future (object filled) frames. From this segmentation a model was created based on edge sets.

Both background subtraction and temporal filtration offers solutions to the problem of motion based image segmentation but they have a number of drawbacks. Neither of these methods succeeds in an environment where there is a dynamically changing background. Example of such situations includes when the camera itself is moving or where areas of non-salient motion are apparent. Background subtraction has the added limitation of requiring a representation of a scene absent of moving objects.

One of the most drawbacks of image difference technique in the detection of moving objects is that it can only capture moving regions with large image frame difference. However, a region can have a small Im_{Diff} even if it is the projection of a moving object due to the aperture problem [15].

$$Im_{Diff} = Im_N - Im_{N-1}, \quad (1)$$

Where: Im_N is the current frame, Im_{N-1} is the previous frame, and Im_{Diff} is the difference frame between the current frame and the previous Frame.

K. Chang and S. Lai proposed an object contour tracking algorithm based on particle filter framework. It only need an initial contour at the first frame and then the object models and the prediction matrix are constructed online from the previous contour tracking results automatically. This proposed algorithm builds two online models for the tracked object, the first gets the shape model and the other gets the grayscale histogram model. The grayscale histogram simply records the grayscale information inside the object contour region. Each of these two models is represented by a mean vector and several principle components, which are adaptively computed with the incremental singular value decomposition technique [16]. E. Trucco, K. Plakas introduce a concise introduction to video tracking in computer vision, including design requirements and a review of techniques from simple window tracking to tracking complex, deformable object by learning models of shape and dynamics [17].

Video tracking can be defined an action to estimate the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to tracked objects in different frames of a video [18]. Obtaining the correct track information of the moving objects is crucial for subsequent actions, like event modeling and activity recognition. For this purpose, many different types of tracking algorithms have been proposed [19, 20]. Most of these algorithms can be listed under the following 4 different groups of video object tracking algorithms:

- 1- Model-based video object tracking,
- 2- Region-based video object tracking,
- 3- Contour-based video object tracking, and
- 4- Feature-based video object tracking.

3. Desirable System Features and Algorithms Necessary for Successful System

3.1 Desirable System Features

The system should be designed with the following general performance measures in minds:

- 1- Ability for real-time operation in complex scenes.
- 2- Adaptability to time-varying target and (slowly varying) background parameters.
- 3- Minimum probability of loss of target (LOT), according to criterion:

$$\min\{E[(B-b)^2]\}, \quad (2)$$

where: B is the actual target location.

b is the estimated target location get from the tracking system.

3.2 Algorithms Necessary for Successful System

The minimum Algorithms necessary for a successful system may be Sub-divided into four parts:

- 1- A target/background (T/B) separation or segmentation algorithm, which can segments the frame by classifying pixels (or groups of pixels) as members of either the target or background sets.
- 2- A tracking filter, to minimize the effects of noisy data which produce an inexact T/B separation that will effect on the estimated target location.
- 3- The used algorithm, which processes information from the just-segmented frame as well as memory information to generate raw estimates of the target centroid (target center).
- 4- An overall system control algorithm, to make the major system automatic decisions, and supervise algorithm interaction.

4. System Description and the Proposed Algorithm

4.1 System Description

Platform Description:

The Platform that used in experiments is:

- 1- Pc computer with capabilities:
 - (i) CPU: Intel Core2Due 1.7 GHz.
 - (ii) 2 Giga byte Ram.
- 2- Web cam with resolution 640 x 480 pixels, and frame rate 25 frame/sec.
- 3- Visual C++ 2008 that used in the implementing phase of the proposed algorithm.
- 4- Matlab 2007 that used in the testing phase of the proposed algorithm.

Input Video Description:

We used for experimental results of the proposed real time video tracking Algorithm a Real Sequence capture by the web cam. For simplicity to trying our proposed algorithm we get the Real sequence of a prototype airplane with simple background. After this sequence we use many sequence of moving target with more texture and real background such as the car sequence that we use in our experiments.

4.2 The Proposed Video Tracking Algorithm

The proposed Real Time Video Tracking Algorithm that we applied depends on the feature extraction of the target. The algorithm description can subdivided into the following steps:

- 1- First, the Algorithm starting with the “Search Mode Module” using frame difference technique to sense the motion in the sequence of the frame. We sense the target motion automatically with (Temporal Filtration).
- 2- Second, after the algorithm senses the motion and indicates that is not a false alarm, it enters the “Tracking Module” (that will be identifying the target and tracking it).
- 3- Third, the algorithm getting the target data, and if the target never lost the algorithm still getting the data, but if the target lost during the number of frames exceeds 5 frames, the algorithm return to the search mode module again.
- 4- Fourth, if the number of target lost frames not exceeds 5 frames the algorithm use a predictor to try to predict the location of the target as in figure 1.
- 5- If the target is still in the field of view of the camera, and the tracking system sense it we use the (Region Growing Technique) to get the whole target by getting the statistical features of the target with the Region Growing Technique.
- 6- After that, we calculate the statistical features for eight directions (East, West, North, South, Northeast, Northwest, Southeast and Southwest) from the center of the moving target in the past frame.
- 7- Calculating the statistical features for eight directions (as in figure 2) of the target center around the predicted centers in the current frame.
- 8- Computing the distance between the statistical features for eight directions of the old center in the past frame and the set of the statistical features for eight directions of the predicted centers in the current frame.
- 9- Get the center that has the minimum distance and minimum difference to be the past center of the target and it will be the new current center.

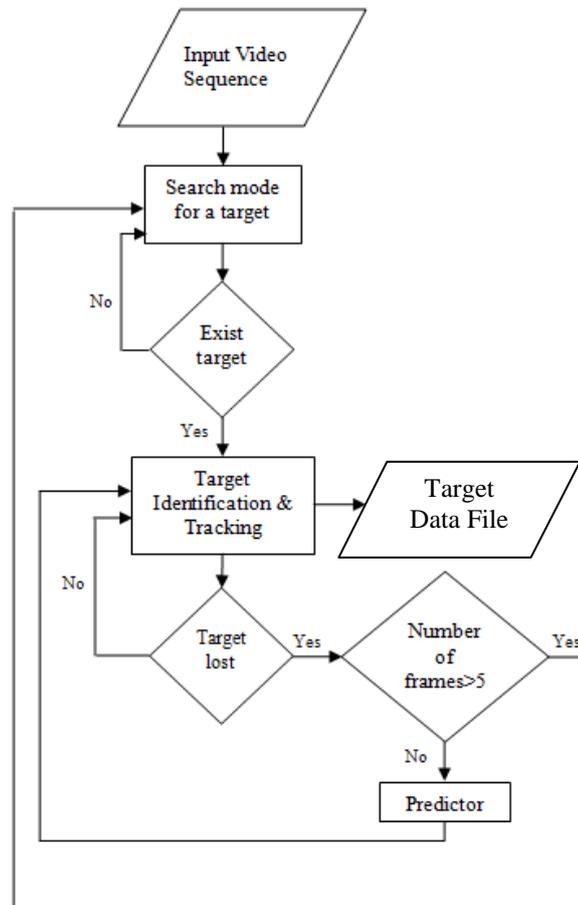


Fig. 1 Proposed Target Tracking Algorithm

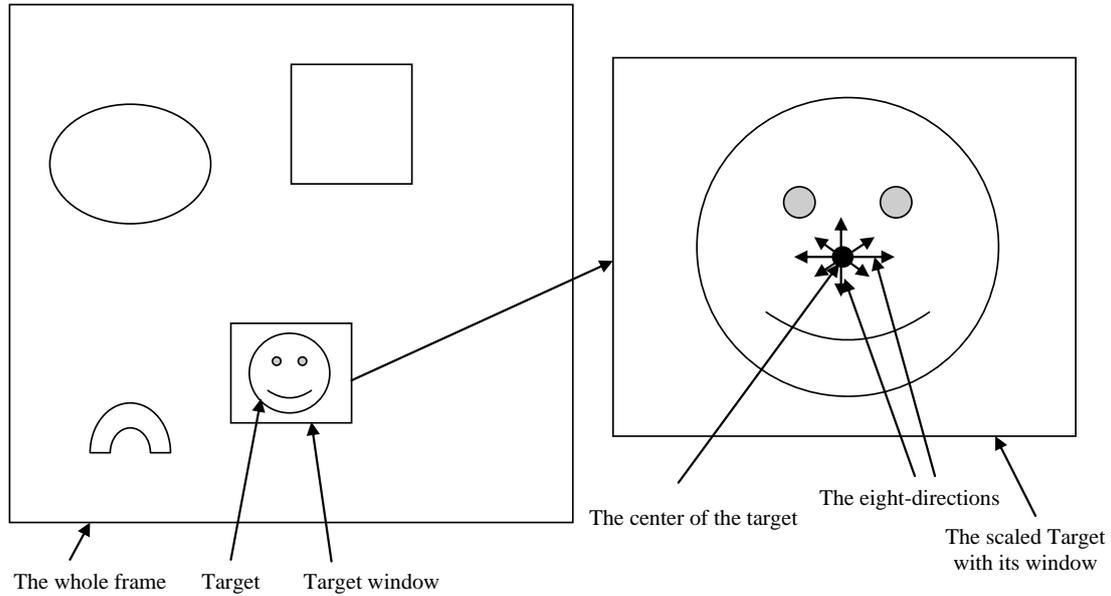


Fig. 2 Getting statistical features of the tracked target in the eight-direction

5. Experimental Results and Comparison for the Proposed Algorithm and the Background Subtraction Algorithm

We used many real video sequences for testing the proposed video tracking algorithm we discuss only four video sequences and compared the proposed algorithm with temporal subtraction algorithm. We divided the testing video sequences into two groups. In the first group we use a recorded video sequences to compare the measured target position with an exact target position to plot an error curves. In the second group we use an online video sequences (not recorded) from the camera direct to the algorithm to test the ability of the proposed algorithm for using in real-time applications.

How can we measure the exact target position?

We can measure the exact target position using mouse pointer with each frame in the sequence and click in the center of the target to get the x-position and the y-position of the target center. For each frame in the video sequence we measure the target position 5 times and get the mean of the target position at this frame to be more accurate.

How can we compute the error in the x-position & y-position?

$$\xi_x (\text{Error in x-position}) = \text{Exact Target Position in (x)} - \text{Measured Target Position in (x)}. \quad (3)$$

$$\xi_y (\text{Error in y-position}) = \text{Exact Target Position in (y)} - \text{Measured Target Position in (y)}. \quad (4)$$

5.1 Experimental Results by the First Group of Video Sequences

We used the first group that we recorded for an airplane toy and a car toy and compare the exact position of the target with the measured position of the target using the proposed algorithm and the temporal filtration algorithm. We get a portion of the trajectory of the target from the whole video sequence between 250 frames for the first video sequence (airplane1 video sequence) and 280 frames for the second video sequence (car1 video sequence).

Figure 3 illustrates a sample of the detection results from the “airplane1” video sequence using the proposed algorithm and, also the following figure 4 illustrates a sample of the

detection results from the “airplane1” video sequence using the Temporal Filtration algorithm.



Fig. 3 Sample of the detection results on the “airplane1” video sequence using the proposed algorithm



Fig. 4 Sample of the detection results on the “airplane1” video sequence using the Temporal Filtration algorithm

Figures 5.a and 5.b illustrate the exact and measured trajectory by the proposed algorithm in the X-Position and the Y-Position. Figures 5.c and 5.d illustrate the corresponding error in the trajectory in the X-Position and the Y-Position.

Also figure 6.a and 6.b illustrates the exact and measured trajectory by the temporal filtration algorithm in the X and Y positions. Figure 6.c and 6.d illustrates the corresponding error in the trajectory in the 2 position X and Y positions. From the 2 figures (figure 5 and figure 6) we noted that:

1. The temporal filtration algorithm fail to detect the target in many frames (i.e The error occurred by the temporal filtration is more than the error occurred by the proposed algorithm) because that algorithm depend upon the frame difference and thresholding techniques, and these techniques are sensitive to noise between pixels that produce from the camera or any light change.

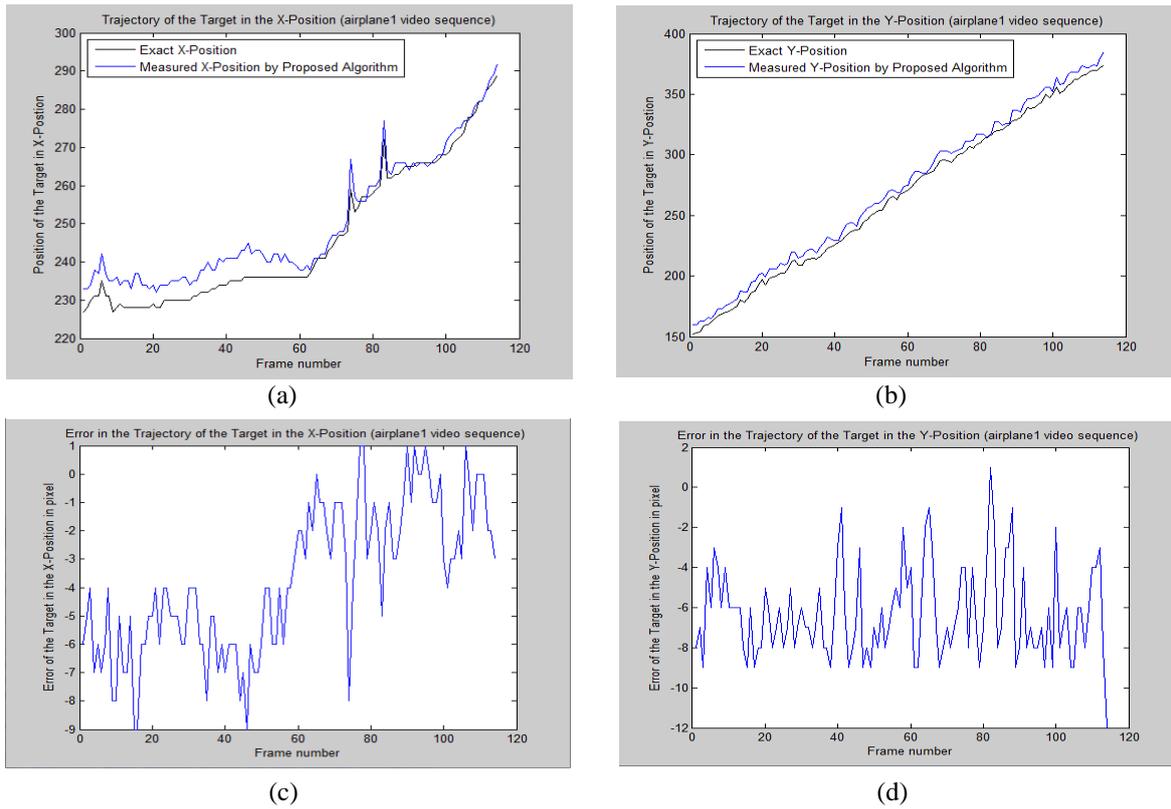


Fig. 5 A portion of the trajectory of the airplane1 using the proposed

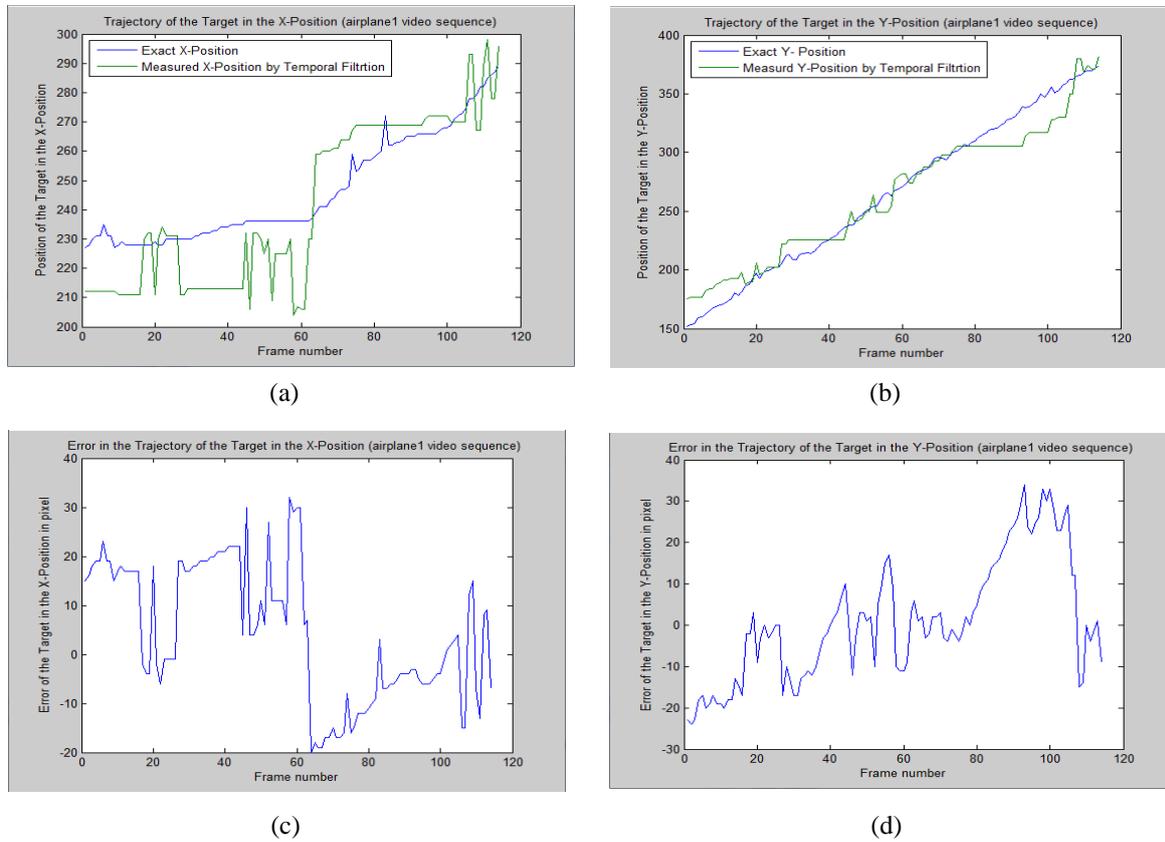


Fig. 6 A portion of the trajectory of the airplane1 using the Temporal Filtration algorithm

2. We noted that from figure 5.a there exist 2 peaks at frame number 73 and frame number 83 these two peaks not an error but it is a high change in the X-Position of the exact trajectory and the proposed algorithm can detect the target at those 2 frames.
3. And also we observed that in the figure 6.a the temporal filtration algorithm can detect the target at the 2 frames correctly because the high change produce a high frame difference that will produce a high detectability for the temporal filtration algorithm.

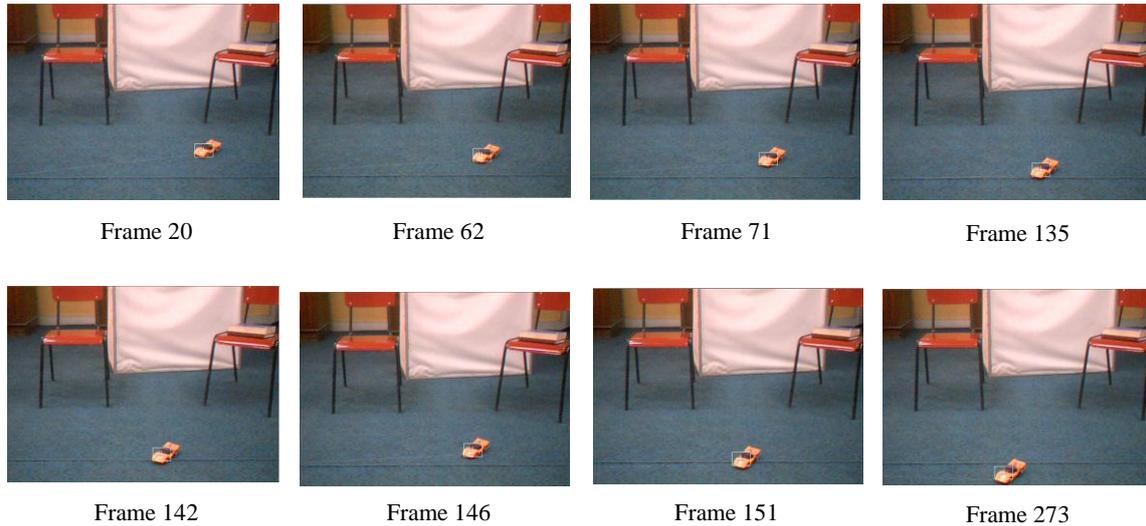


Fig. 7 Sample of the detection results on the “car1” video sequence using the proposed algorithm

Figure 7 illustrates a sample of the detection results from the “car1” video sequence using the proposed algorithm and, also figure 8 illustrates a sample of the detection results from the “car1” video sequence using the Temporal Filtration algorithm.

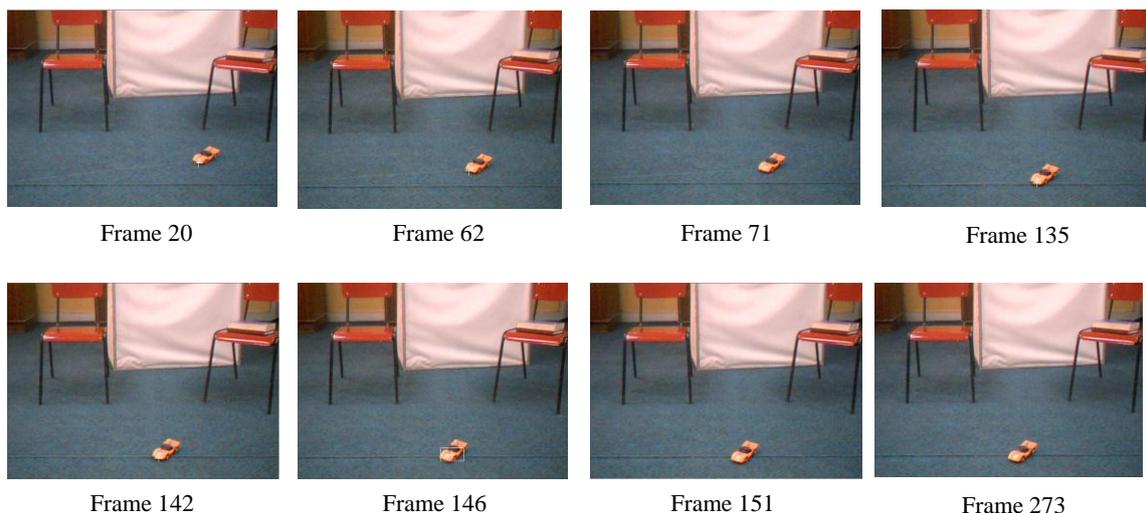


Fig. 8 Sample of the detection results on the “car1” video sequence using the Temporal Filtration algorithm

Figure 9.a and 9.b illustrates the exact and measured trajectory by the proposed algorithm in the X-Position and the Y-Position. Figure 9.c and 9.d illustrates the corresponding error in the trajectory in the X-Position and the Y-Position.

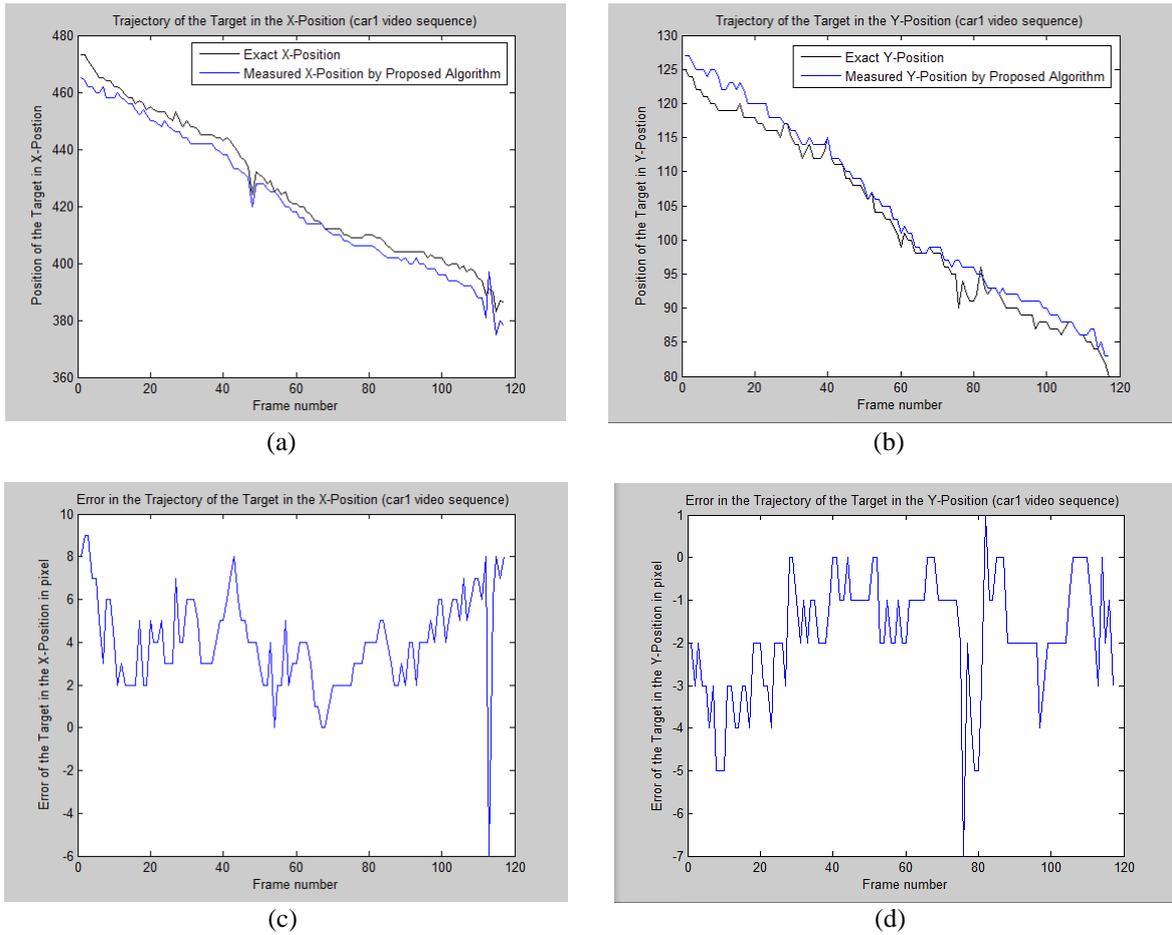


Fig. 9 A portion of the trajectory of the car1 using the proposed algorithm

Also the following figure 10.a and 10.b illustrates the exact and measured trajectory by the temporal filtration algorithm in the X and Y positions. Figure 10.c and 10.d illustrates the corresponding error in the trajectory in the 2 position X and Y positions.

1. The temporal filtration algorithm fail to detect the target in many frames (i.e The error occurred by the temporal filtration is more than the error occurred by the proposed algorithm) because that algorithm depend upon the frame difference and thresholding techniques, and these techniques are sensitive to noise between pixels that produce from the camera or any light change.
2. We noted that from figure 10.a and figure 10.b the correct detectability of the temporal filtration algorithm is low in the “car1.avi” sequence compared by the “airplane1.avi” sequence because the “airplane1.avi” is a simple sequence and the background is simple but the “car1.avi” is complex scene and has more object in the frame (i.e. the background is complex) that means more noise to the temporal filtration algorithm.
3. The second reason for that fail in the detection of the target by the temporal filtration algorithm is the low motion of the target (i.e. the frame most has high changes to detect the target and if the target stopped suddenly this algorithm can’t detect any target), but the proposed algorithm can detect the target even the target is stopped because it depend upon the extraction of the feature of the target.

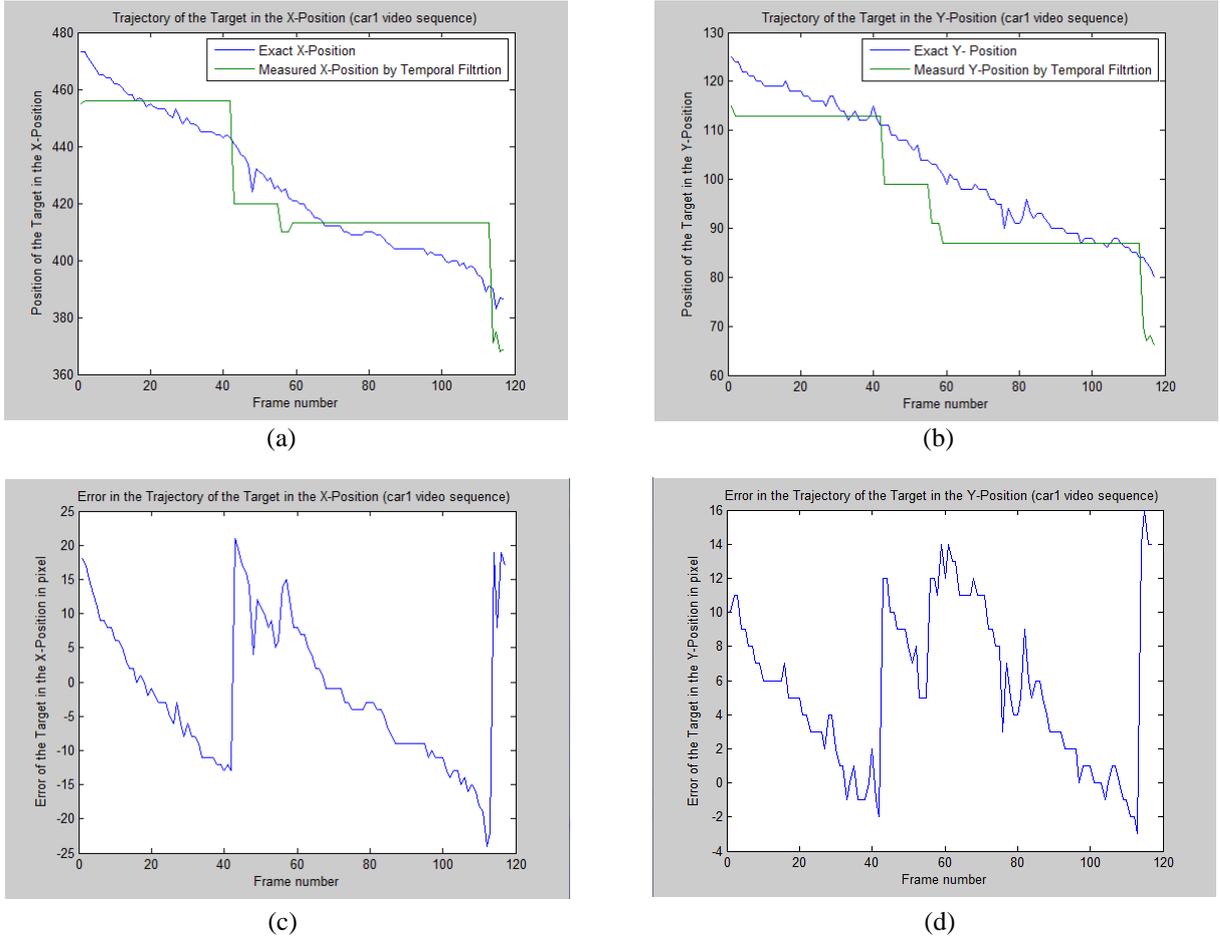


Fig. 10 A portion of the trajectory of the car1 using the Temporal Filtration algorithm

5.2 Experimental Results by the Second Group of Video Sequences (Real-Time)

We used for the second group online video sequences (not recorded) from the camera direct to the algorithm for the experimental results of the proposed video tracking algorithm to test the capability of the proposed algorithm for using in the real time application, we try to simulate the same sequences for at the same environmental condition such as lighting condition and camera position and approximately simulate the same trajectory of the target for the purpose of testing our proposed algorithm with the temporal filtration algorithm.

We choose two real video sequences and we discuss them:

- 1- The airplane2 video sequence.
- 2- The car2 video sequence.

Figure 11 illustrates some of the frames of the “airplane2” sequence after tracking; a, b, c, and d are the tracked frames after applying the temporal filtration & d, e, f, and g are the tracked frames after applying our proposed real-time algorithm.

Also figure 12 illustrates some of the frames the “car2” sequence after tracking a, b, c, and d are the tracked frames after applying the temporal filtration & d, e, f, and g are the tracked frames after applying our proposed real-time algorithm.

From figure 11 we noted that the temporal filtration algorithm that applied on the “airplane2” video sequence loss the target data as in figure 11.c, because the target has sudden stop and the algorithm does not response any change in the frame difference, and loss the correct location of the target as in figure 11.d because the change in frame can’t get a noise from light change. But when we applied our proposed algorithm we get the target locations in the whole sequence and did not loss any location on any frame even if the target stopped as in figure 11.e, f, g, h.

From figure 12 we noted that the temporal filtration algorithm that applied on the “car2” video sequence loss the target data as in figure 12.a and figure 12.c, because the target has sudden stop and the algorithm does not response any change in the frame difference. But when we applied our proposed algorithm we get the target locations in the whole sequence and did not loss any location on any frame even if the target stopped as in figure 12.e, f, g, h.

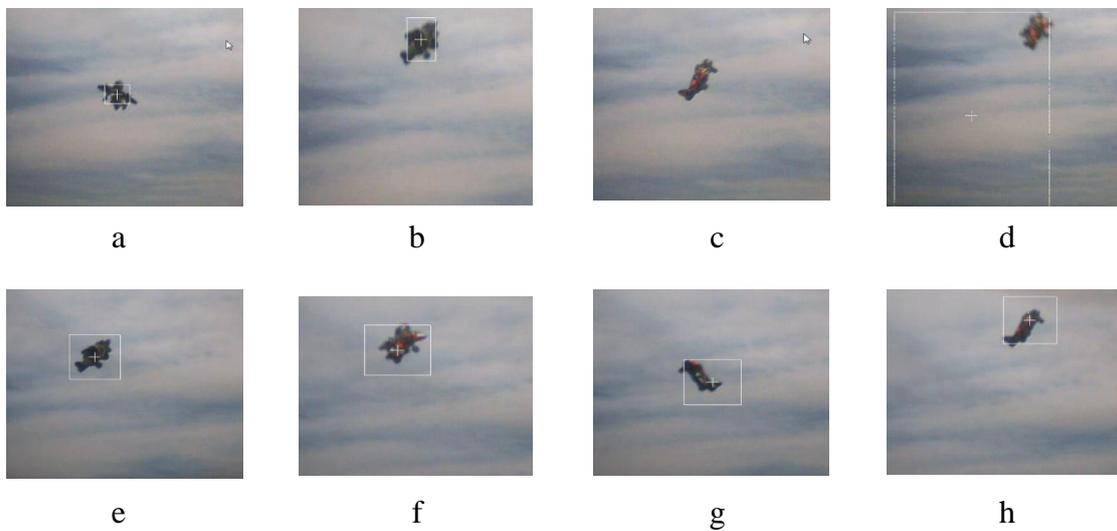


Fig. 11 The airplane2 video sequence

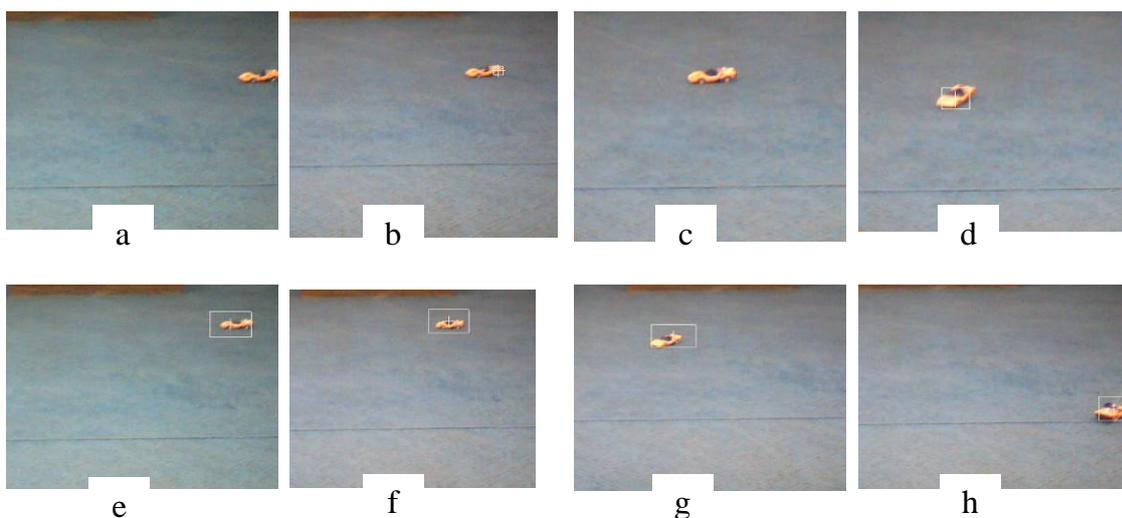


Fig. 12 The car2 video sequence

6. Analysis of the Obtained Results

From the experimental results and figure 3 through figure 12 we found that:

- 1- Temporal filtration is difficult to handle shadow and occlusion.
- 2- Temporal filtration fails to track the position of the object correctly at sudden changes in background or brightness.
- 3- Temporal filtration failed to track the position of the object that suddenly stopped.
- 4- Our proposed real-time algorithm successfully tracks an object in real-time application with high frame rate (25 frames/sec, with resolution 640 x 480).
- 5- Our proposed algorithm can detect and track any object within a frame and does not need to produce a background frame, and does not need to know any feature about the object. But it will produce a feature table for the object that will be tracked and tracking it in the whole sequence frames.
- 6- This algorithm is a real-time object tracking based on dynamic feature extraction grouping with temporal subtraction module in the searching mode only.
- 7- Our proposed algorithm saves computation time that is needed for tracking an object over the sequence of frames by limiting the searching area in the predicted small window located within the center of the object.

The obtained results in this paper showed the robustness of using dynamic feature extraction with the temporal filtration. This research is a part of a project of real-time multi-object tracking.

7. References

- [1] V. Manohar, P. Soundararajan, H. Raju, D. Goldgof, R. Kasturi, J. Garofolo, "Performance Evaluation of Object Detection and Tracking in Video," LNCS, vol. 3852, 2(2006), pp.151-161.
- [2] Y. T. Hsiao, C. L. Chuang, Y. L. Lu, J. A. Jiang, "Robust multiple objects tracking using image segmentation and trajectory estimation scheme in video frames", *Image and Vision Computing*, vol. 24, 10(2006), pp. 1123-1136.
- [3] T. Ellis, "Performance metrics and methods for tracking in surveillance", 3rd IEEE Workshop on PETS, Copenhagen, Denmark (2002), pp. 26-31.
- [4] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, "Color-based probabilistic tracking", *Conference on Computer Vision*, LNCS, vol. 2350 (2002), pp. 661-675.
- [5] T. Schoenemann, D. Cremers, "Near Real-Time Motion Segmentation Using Graph Cuts", *Springer, DAGM 2006*, LNCS 4174, 2006, pp. 455-464.
- [6] R. Polana and R. Nelson, "Low level recognition of human motion." *Proceedings IEEE Workshop Motion of Non-Rigid and Articulated Objects*, Austin, TX, 1994, pp. 77-82.
- [7] Burky B. ÖRTEN, "Moving Object Identification and Event Recognition in Video Surveillance Systems", Master of Science, The Graduate School of Natural and Applied Sciences of Middle East Technical University, July 2005.
- [8] Javed Ahmed, M. N. Jafri, J. Ahmed, M. I. Khan, "Design and Implementation of a Neural Network for Real-Time Object Tracking", *World Academy of Science, Engineering and Technology*, June 2005.
- [9] M. Yang and N. Ahuja, "Detecting Human Faces in Color Images", *Proceedings IEEE International Conference on Image Processing*, IEEE Computer Soc. Press, Los Alamos, Calif., pp. 127-139, 1998.
- [10] David N. McKinnon, "Multiple Object Tracking in Real-Time", Undergraduate Thesis, Univ. Queensland, St Lucia, Dept. Computer Science and Electrical Engineering, 1999.

- [11] Daniel R. Corbett, "Multiple Object Tracking in Real-Time", Undergraduate Thesis, Univ. Queensland, St Lucia, Dept. Computer Science and Electrical Engineering, 2000.
- [12] J. Vass, K. Palaniappan, X. Zhuang, "Automatic Spatio-Temporal Video Sequence Segmentation", Proc. IEEE International Conference on Image Processing V3, IEEE Computer Soc. Press, Los Alamos, Calif., pp.958-962, 1998.
- [13] A. M. Sallam, M. Sharrawy, O. Elmouafy, R Elbardany, A. Fahmy, "Object Based Video Coding Algorithm", Proceedings of the 7th International Conference on Electrical Engineering, ICEENG 2010, May 2010.
- [14] Robert Andrews, "Multiple Object Tracking in Real-Time", Undergraduate Thesis, Univ. Queensland, St Lucia, Dept. Computer Science and Electrical Engineering, 1999.
- [15] Berthold. K. P. Horn, "Robot Vision", Mc Graw-Hill Book Company, New York, 1986.
- [16] K. Chang, S. Lai, "Adaptive Object Tracking with Online Statistical Model Update", Springer, ACCV 2006, LNCS 3852, 2006, pp. 363-372.
- [17] E. Trucco, K. Plakas, "Video Tracking: A Concise Survey", IEEE Journal of Oceanic Engineering, Vol. 31, No. 2, April 2006.
- [18] Duane R. Edington, Ishbel Kerkez, Danelle E. Cline, Dorothy Oliver, Marc Aurelio Ranzato, Pietro Perona, "Project of Detecting , Tracking, and Classifying Animalsin Under water video," New York university, 2002.
- [19] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, no. 7, pp. 780-785, July 1997.
- [20] Hu, W., Tan T., Wang L., Maybank S., "A Survey on Visual Surveillance of Object Motion and Behaviours" IEEE Transactions on Systems, Man, and Cybernatics, Vol. 34, no. 3, August 2004.