Journal of Engineering and Development, Vol. 16, No.4, Dec. 2012 ISSN 1813- 7822

Tracking Methods for Multipurpose Video Processing Applications

Dr. Muthana H. Hamd Al Amiri
University of Al-Mustansirya
College of Engineering
Computer and Software Eng. Dep.
Muth700@yahoo.com

Abstract:

Video target tracking is a complex and time consuming process because of huge data that is contained in video stream. In this paper, a real-time software is developed to track any motion in a video zone for surveillance/security applications. First, an interested object is detected, isolated (segmented), and localized among different objects of video frames. Secondly, three tracking procedures are applied to find the efficient one that satisfies the real-time requirements for tracking the target with minimum response time. The tracking model consists of two main parts, the web camera which represents the video acquisition device that monitors any motion in a dedicated zone. The second part is the software that controls and processes the video stream; it captures the frames and tracks the new target center coordinates. The center values that will be found by tracking algorithms are very essential for other video/image applications like radar system in military applications, road traffic control applications, or kidney stone ultrasound in the medical applications. The experimental results found that the response time and targeting accuracy of the three methods are very close, the maximum difference in a response time was 0.0421 sec for threshold Θ = 0.039.

الخلاصة

تقنية تتبع هدف فيديوي من العمليات المعقده وتستغرق وقت بسبب الكميات الهائلـه من البيانـات الموجوده فـي الحزمـه الفيديويه. في هذا البحث, تم تطوير برنامج زمن حقيقـي لتتبـع ومراقبـة مسار اي هدف متحرك ضمن المنطقـه المحدده. أولا": يتم كشف الهدف ثم تمييزه (عزلـه) و تحديد موقعه عن بقيـة الاجسام والخلفيات الموجوده فـي سلسلة الصور الفيديويه. ثانيا": تم أعتماد ثلاثة خوارزميات لتتبع الهدف وايجاد مركزه والغرض هو اختيار أكفأ واحده تحاكي الزمن الحقيقي في تتبع الهدف باقل زمن استجابه. يتكون نموذج تتبع الهدف من جزئين اساسيين, الكاميرا لطلب الحزمـه الفيديويه ومجموعة البرامج والسواقات التي تقوم بتعريف المدخلات والمخرجات والسيطره على تنفيذ وسير العمليـات. يكمن استخدام هذا النموذج في تطبيقات الرادار وانظمة المرور والتطبيقات الطبيه مثل حصى الكلى. النتـائج المختبريـه اظهرت تقارب نتائج الخوارزميات الثلاث. ان أعظم زمن اختلاف كان 0.0421 ثانيه لقيمة عتبه تساوي 0.039

Keywords: Video tracking, tracking algorithm, targeting

1. Introduction

Humans are very interested with visual skills; they can identify a face in an instant and process a large amount of visual information very quickly. Image processing can deal with changing scenes, for example, an image is a single picture which represents something. It may be a picture of a person, people or animals, an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging [1-3]. Digital image processing may includes the processes of acquiring an image of the area containing interested object, pre-processing that image, extracting (segmenting) the desired object, and recognizing this object. Generally, image means a still picture that does not change with time, whereas a video evolves with time and generally contains moving and/or changing objects. The amazing diversity of image processing/analysis applications is explained in Fig. 1 [4][5]. Digital image are usually obtained from video stream using sampling and quantization techniques, which means that once the image/video signal is sensed, it must be converted into a computer-readable, digital format (A/D conversion), although "direct digital" systems are becoming more prevalent. Likewise, digital images are viewed using diverse display media, included digital printers, computer monitors, and digital projection devices, see Fig. 2. In this paper, the general architecture of the system proposal is described in section 3. The video acquisition device is explained in section 4. Sections 5 and 6 present three tracking procedures with different examples. Experimental results are compared and tabulated in section 7. The last section, 8 is the paper conclusion.

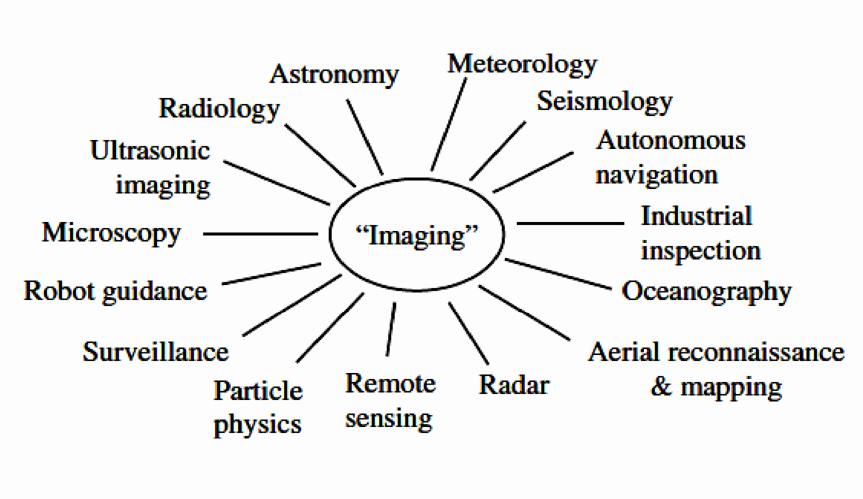


Fig. 1. Video/Image processing applications

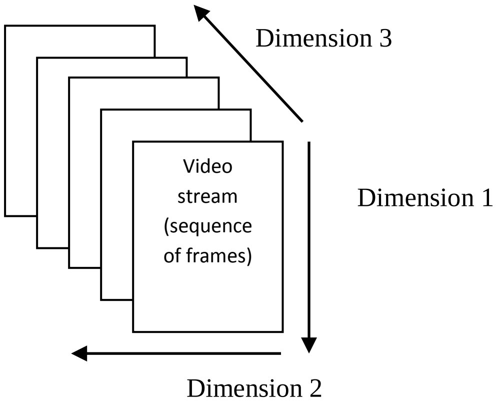


Fig.2 Image/video dimensionality

2. Frame-based Processing

Frame-based processing is used to accelerate both real-time systems and simulations where frame-based data is a common format in real-time systems. Data acquisition hardware often operates by accumulating a large number of signal samples at a high rate, and propagating these samples to the real-time system as a block of data. The frame-based throughput rate is therefore many times higher than the sample-based alternative; the efficiency of the system will be maximized by distributing the fixed process overhead across many samples. The "fast" frame acquisition is suspended by "slow" interrupt service routine (ISR) processes after each frame is acquired, rather than after each individual sample. It's important to note that frame-based processing introduces a certain amount of latency into a process due to the inherent lag in buffering the initial frame. In many instances, however, it is possible to select frame sizes that improve throughput without creating unacceptable latencies [4].

3. The Proposed Model

The target movement in a video stream requires a tracking algorithm to analyse the sequential video frames and output the target motion. Most of these algorithms use tracking system components like: target detection, target isolation, filtering, and data association. Some of those algorithms are:

- Blob tracking: segments the object interior
- Kernel-based tracking, also called mean-shift tracking, It represents an effective localization procedure by maximizing the object similarity measurment.
- Contour tracking: detects the object boundary

Locating and tracking the target object successfully is dependent on the algorithm. For example, using blob tracking is useful for identifying human movement because a person's profile changes dynamically. The general block diagram of our tracking system is described in Fig. 3 [8][9]. Mainly it consists of two components: the software which is responsible for:

- Video stream acquiring
- Object detection and segmentation
- Center(s) computation
- Tracking control

The second component is a video acquisition device which is responsible for zone monitoring.

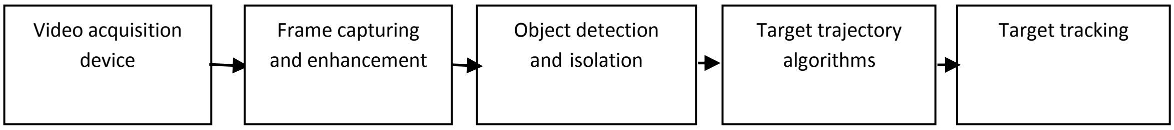

Video acquisition device
Frame capturing and enhancement
Object detection and isolation
Target trajectory algorithms
Target tracking

Fig.3 Tracking system model

4. Video Stream Acquiring

A Web camera has been selected to monitor the secure zone experimentally. The input video stream covers any motion in a distance of 30 meter long and 53.12^{0} forward zone angel, see Fig. 4. The surveillance zone can be wider and longer if a special video input device is used rather than a Web camera. However object detection and tracking depends on quality of the input video stream and hence the video input device type, so the proposed model is designed to track any interested object even for radar or satellite applications that represents so far distance [10-12].

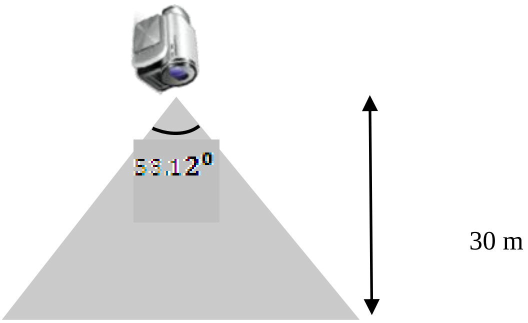


Fig.4 Surveillance zone description

5. Tracking Procedure

Video frame sequences are continuously captured and processed to detect any new object or tracking a target in the surveillance zone. Tracking procedure includes:

(1) Motion and edge detection

(2) Target trajectory computations,

(3) Response time calculation

Image enhancement operations like; noise removal, smoothing, and thresholding will precede the segmentation. The trajectory algorithms are continued track the target by finding its (x,y) center. The binary threshold value is adaptable, that means it depends on the brightness of the frame. It takes high values, close to 1 for high light environment and small values close to 0 for darkness one.

6. Trajectory Algorithms

Three different procedures have been applied to compute the target center and hence its trajectory. Next sections will demonstrate these algorithms in detail.

6.1 Four Crossing Method

The next step after the object has been detected and isolated, is to find its center values as it became a target. A Plumb Line Centroid method is modified to fit the algorithm demands. The centroid of any 2D shape is the intersection of all straight lines that divide this shape into two parts of equal moment about the line. The centroid of a 2D object may be determined, experimentally, by using a plumb line to find the center of mass of a thin object. The position of the plumb line is traced on the body. The experiment is repeated at different points of the object. The intersection of the two lines is the centroid of the shape. For example the four straight lines those intersect the target shown in Fig.5 having two intersection points. The first (r_1, c_1) is (4, 7) and it is resulted from the two intersection lines, $\vec{A}$ X $\vec{B}$. The second (r_2, c_2) is (7, 5) and it is resulted from the two intersection lines, $\vec{C}$ X $\vec{D}$. Where $\vec{A}$, $\vec{B}$, $\vec{C}$ and $\vec{D}$ are started from the first edge pixels of the top, right, bottom, and left respectively. So the target center will approximately be

$$\text{Target center} = \frac{abs\left[(r_2, c_2) - \left(r_{1,} c_1 \right)\right]}{2} + [(r]_{1,} c_2) \qquad (1)$$

$$= (1.5, 1) + (4, 5) \cong (5,6) \quad \text{or}$$

[(4,7) + (7,5)] / 2 → (5,6)

Fig.6 clearly explained the target center and its edges. This method gives good results for different target shapes; however for some shapes like a crescent, it is better to use different methods as discussed in the next section.

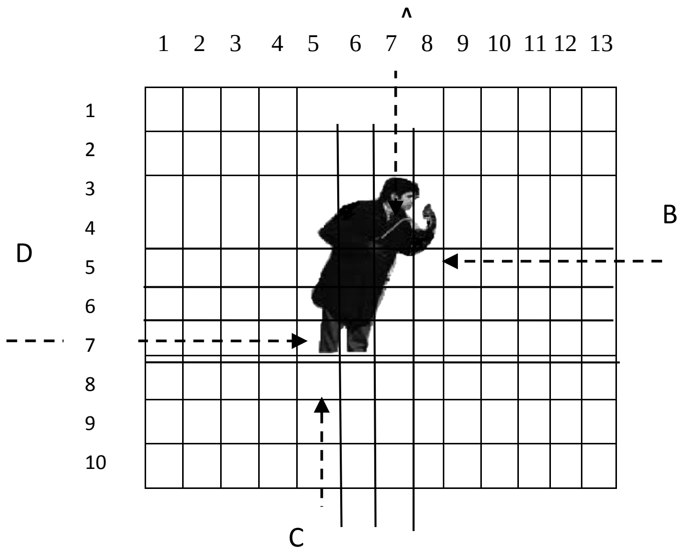


Fig. 5 Target center using four lines intersection

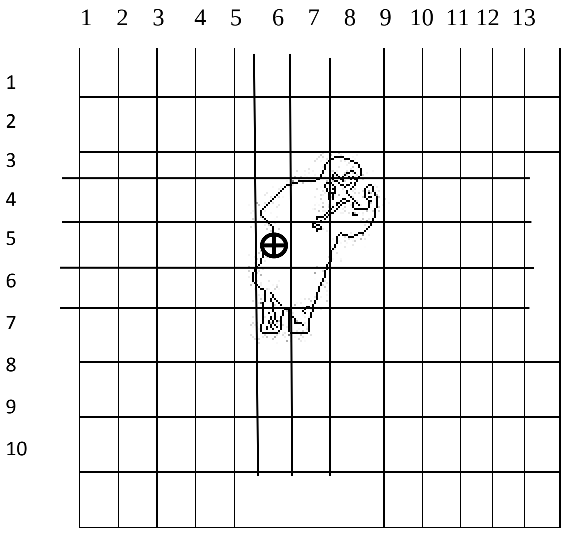


Fig 6 Taget center at pixel (5,6)

6.2 Clusters Mean Center

The MATLAB centroid function is modified to find the mean center of target clusters. For this method, the object may be processed as group of clusters, so the mean center of the two vectors $\vec{x}$ and $\vec{y}$ will represent the mean center of the target as illustrated in Fig.7

Target center = Mean ($\vec{x}$, $\vec{y}$) (2)

Mean ($\vec{x}$) = $\vec{x}$ / 6 = (3+4+5+5+6+7) / 6 = 5

Mean ($\vec{y}$) = $\vec{y}$ /6 = (5+5+6+7+7+8) / 6 = 6

Target center = (5,6)

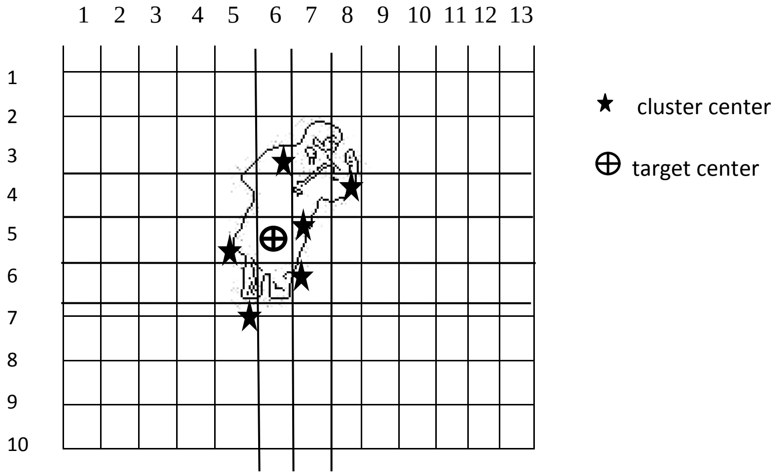


Fig. 7 Mean center of target using Clusters Mean Center

6.3 Longest Vector Center

A new procedure named *Longest Vector Center* is adopted and developed to find a valid and guaranteed target center. With this procedure, the longest two target vectors are computed amongst target rows and columns. The center of the longest row and column will be selected as a center to track the target. This method is more efficient for irregular shapes where the first method may produce invalid center value. Example of this shape is the crescent as explained in Fig. 8. The center values found by the first method will be (4, 7) and it lies out of the target boundary depending on the following calculation:

abs [(2,9) – (7,4)]/2 + (2,5)

= (2, 2) + (2,5)

= (4,7)

With this method, the column vector $\overrightarrow{AB}$ contains the pixel pairs [(3,4),(4,4), (5,4), (6,4), (7,4)] and the row vector $\overrightarrow{CD}$ contains the pixel pairs [(2,5), (2,6), (2,7), (2,8),(2,9)] and they are representing the longest row and column vector respectively of the target shown in Fig. 8. So the two center values will be:

$$\text{Target center} = \frac{\text{Max}\left(\frac{\text{row}}{\text{column}}\right)}{2} \qquad (3)$$

Target center-1(max row) = [(7,4) – (3,4)] / 2 + (3,4) = (5,4)
Target center-2 (max column) = [(2,9) – (2,5)] / 2 + (2,5) = (2,7)

The two values are valid and lay on the target body as explained in Fig. 8.

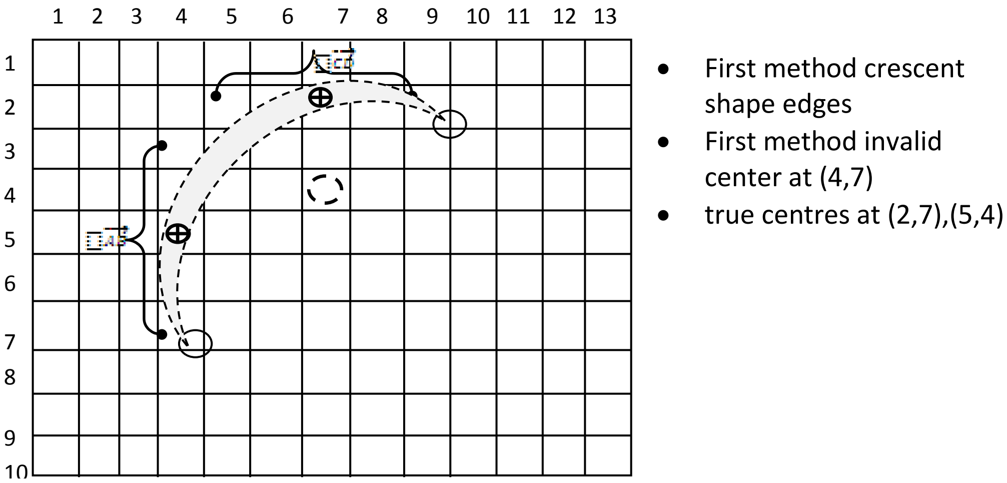

1 2 3 4 5 6 7 8 9 10 11 12 13
1
2
3
4
5
6
7
8
9
10
$\overrightarrow{CD}$
$\overrightarrow{AB}$
• First method crescent shape edges
• First method invalid center at (4,7)
• true centres at (2,7),(5,4)

Fig. 8 Crescent shape center using longest vector method

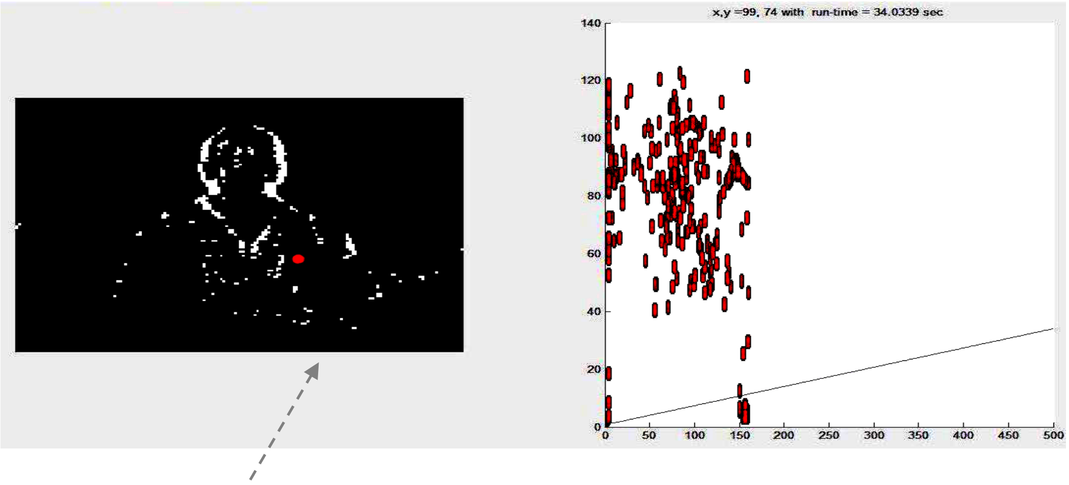
7. Three Methods Results and Comparison

The three methods have been applied and run on 1000 frames of video stream with 30 frames per trigger. The test is done under same conditions of illumination, distance and target appearance to find the best method that satisfies real-time requirements regarding minimum response time and targeting accuracy. The software implementation results are tabulated in table 1. Fig.(9, 10, and 11) are examples of implementation for three different threshold values(θ1=0.029, θ2=0.039, and θ3=0.049). They contain a binary target image, process time vs. frame pair, and target trajectory. The execution times for the three methods are too close (34.xxx sec.) and the minimum one is (34.0339 sec., θ1=0.029) for four crossing method, while the longest one is (34.0787 sec., θ1=0.029) for longest vector center method. From table

1 results, it's clear that for every column (threshold) the four crossing method has always a minimum run time, while the longest vector method takes longest run time.

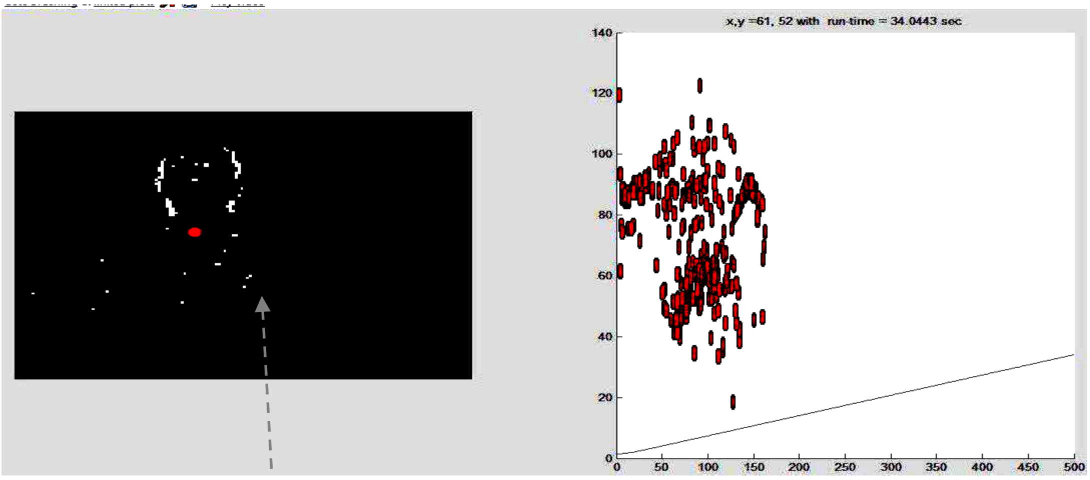
Table 1. Three methods experimental results

Method	Run time (sec) for 1000 frames			Target center (x,y)		
	Θ_1=0.029	Θ_2=0.039	Θ_3=0.049	Θ_1=0.029	Θ_2=0.039	Θ_3=0.049
Four crossing	34.0339	34.0366	34.0366	(99,74)	(65,54)	(104,49)
Clusters mean center	34.0352	34.0443	34.052	(99,68)	(61,52)	(88,60)
Longest vector center	34.0445	34.0787	34.0776	(100,60)	(100,60)	(60,30)



Target center

Fig. 9 Target image, target trajectory, and process time vs. frame pair for four crossing Method



Target center

Fig. 10 Target image, target trajectory, process time vs. frame pair for clusters mean center method

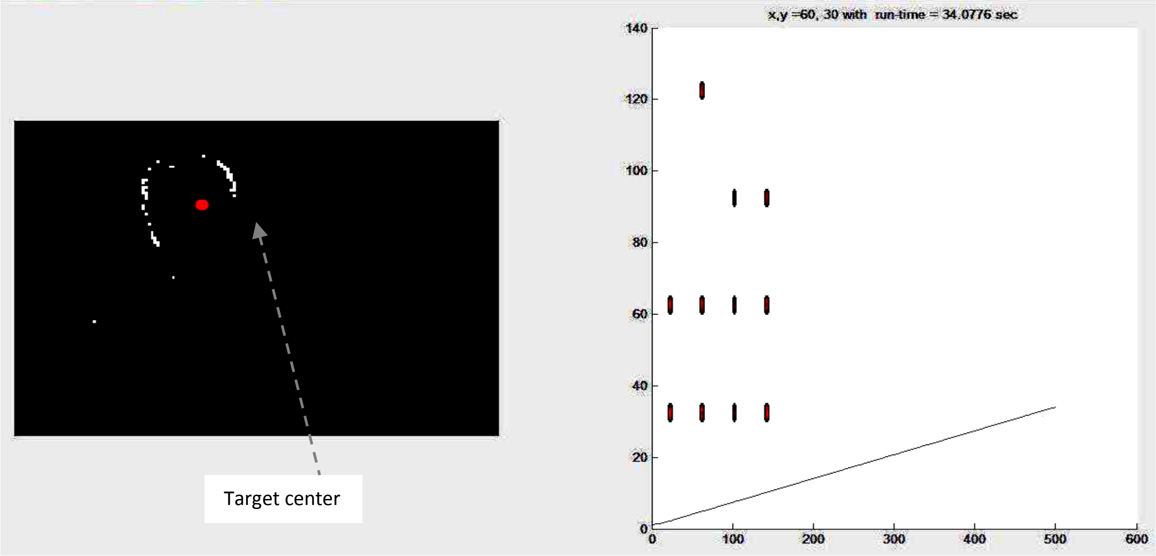


Fig. 11 Target image, target trajectory, process time vs. frame pair for longest vector Method

8. Conclusion

Three tracking methods have been implemented and compared for video stream of a built in Web camera. For 1000 sequential video frames and three different threshold values, the Four Crossing method has had the shortest response time, while the Longest Vector method takes longer time. An irregular target center (x,y) like a thin crescent shape could be granted with Longest Vector Method. From table 1, the tracking time goes longer as threshold becomes higher for all methods. These tracking methods are developed to simulate the real-time tracking procedures regarding response time and targeting accuracy. They can be used for many applications ranging from radar systems to video surveillance/security applications.

References

[1]Alasdair McAndrew, ”**An Introduction to Digital Image Processing With MATLAB**”, Notes for SCM2511 Image Processing, Victoria University of Technology, 2004

[2]WILLIAM K. PRATT, “**Digital Image Processing**”, A John Wiley & Sons, Inc., Publication 4th Edition, 2007

[3]Xiaofei He, Shuicheng Yan, “**Face Recognition Using Laplacianfaces**”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, no. 3, March, 2005

[4]Rafael G. Gonzalez, E. Richard Woods, "**Digital Image Processing Using MATLAB**", A John Wiley & Sons, Inc., Publication 4th Edition , 2004

[5]Gérard Blanchet Maurice C., "**Digital Signal and Image Processing using MATLAB**", ISBN 10: 1-905209-13-4, 2006

[6]Md. Mahmudur Rahman, Bipin C. Desai , "**Image Retrieval-Based Decision Support System for Dermatoscopic Images**" Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, 0-7695-2517-1/06, 2006

[7]Christoph Bauer', Fabian J, "**Local Features in Biomedical Image Clusters extracted with Independent Component Analysis**", Proceeding of the Int. Joint Conference on Neural Networks, 26th Aug., 2005

[8]Xiaojing Yuan , "**SVM-based Texture Classification and Application to Early Melanoma Detection**", Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept 3, 2006

[9]Yu Zhou1 Melvyn Smith **"Segmentation of Clinical Lesion Images Using Normalized Cut** ", IEEE conference, 978-1-4244-3610-1/09/, 2009

[10]Cláudio R. Jung, "**Sharpening Dermatological Color Images in the Wavelet Domain**", IEEE journal of Selected Topics in Signal Processing, Vol. 3, No. 1, February 2009

[11]JiangFeng Chen, BaoZong Yuan, "**Face manifold analysis based on LFA features**", Congress on Image and Signal Processing, 978-0-7695-3119-9/08 2008

[12]Xiangyun Qing and Xingyu Wang, "**Face Recognition using Laplacian+OPRA-faces**", Proceedings of the 6th World Congress on Intelligent Control and Automation, China, 2006