Military Technical College

Kobry El-Kobbah,

Cairo, Egypt

8th International Conference on Electrical Engineering

ICEENG 2012

AUTOMATIC ADAPTIVE WINDOW TRACKING BASED ON COLOR PROBABILITY DISTRIBUTION

Wael Mohamed Yousf, Osama Mohamed Elmowafy*, Ibrahim Ali Abdl-Dayem*,*
*Alaa Eldin Mohamed Fahmy**

ABSTRACT

This paper introduces a proposed modified approach of an end-to-end technique for moving targets tracking. The tracking technique is processed on a real-time video stream. The proposed approach is a prolongation of the Continuously Adaptive Mean Shift (CAMShift) algorithm applications. Sever variations in target shape, size and luminosity can be dealt better using this algorithm. Edge detection technique is used to deal with the change in target shape and size. An estimator is used to deal with luminosity changes. A proper size of tracking window is built with minor extra computational overhead. Experimental results show the effectiveness of the proposed algorithm.

Keywords: *Tracking,* CAMShift*, edge detection*

I. INTRODUCTION

Efficient tracking system in surveillance applications is one of the most important and challenging task within the field of computer science. Several proposed techniques are designed for static cameras environments. With moving cameras, moving objects tracking become more difficult and many techniques failed to detect and track the desired targets. Also, the complexity due to noise in images, complex object motion, partial and full object occlusions, complex object shapes, scene illumination changes, and real-time processing requirements should be taken into consideration. The problem becomes more complex when dealing with a specific object in real-time using a moving PTZ camera in order to keep the target within the image.

The CAMShift algorithm [1] was derived from the earlier Mean Shift algorithm [2]. It is a simple but not effective color-based tracking technique. It is considered as basic component for many advanced tracking systems. It can be used in face tracking, video surveillance, video editing, and computer vision based game interfaces. Stable tracking despite appearance changes (e.g., due to lighting or perspective) and partial or full occlusion, and re-detection of lost objects are other problems of standard CAMShift.

II. RELATED WORK

Since its introduction as a technique for face tracking, CAMShift has been subjected to a variety of modifications to accommodate with other tracking applications. Mean Shift is a predecessor of CAMShift that does not update the search window size. The most related works to CAMShift and Mean Shift implementations is introduced.

Kok Bin et al. [3] explained another enhanced CAMShift implementation for face tracking. Pixels in the search window are also weighted as proposed in [4][5]. Applying perceptual grouping to the back-projected probabilities helps to eliminate background noise as explained in [6]. Furthermore, they continuously adapt the target histogram by re-computing it for the current search window if the object was reliably tracked. A similar strategy is followed in [7] yet, the histogram is always updated. This adapts to continuous appearance changes, but fails in case of abrupt changes.

Zhang et al. [8] combined Kalman filtering with CAMShift tracking to avoid converging to local maxima and to enable track recovery after full occlusions. A background-weighted histogram was used to distinguish the target from the background and from other targets.

Robin Hewitt [9] explained OpenCv CAMShift algorithm for face tracking. CAMShift uses color information, but rather than relying on a single color, it tracks a combination of color. Since it tracks by color, it can follow a face through orientation changes that the Haar detector can't handle, but fails in automatic tuning parameter, and their color values are Unstable and these pixels contribute noise that interferes with tracking.

Karan Gupta, Anjali V. Kulkarni [10] explained the dynamic template matching and frame differencing implement for single object tracking system using a single pan, tilt camera, but fails in sudden change in the direction of motion.

All of the approaches that are summarized above extend CAMShift or Mean Shift in the one or the other way. However, none of them took adequate measures to enhance tracking in all aspects. Severe appearance changes (caused by changes of illumination or perspective) are not addressed in [8], re-detection after track losses is not supported in [3] [7][10], occlusions are not resolved in [3][4], and self tuning and de-noising are not treated in [9]. We address all of these points latter in details.

III. THE CAMSHIFT ALGORITHM

The principle of the CAMShift algorithm is given in [8]. Each image of the sequence is converted into a probability distribution image relative to the histogram of the object to be tracked. From this image, the centre and the size of the object are given based on the CAMShift algorithm. These new centre and size are employed to place the search window in the next image. This process is repeated for a continuous target tracking in the video sequence. The algorithm of CAMShift employs a 2D probability distribution image produced from a back-projection of the target histogram with the processed image. The CAMShift algorithm calls the Mean Shift for one time only to calculate the target centre of the probability distributed image. It is a matter of finding a rectangle presenting the same moments as those measured on the image probability. These parameters are given from the first and second moments [11].

IV. PROPOSED APPROACH

In this paper, a real-time target tracking technique using real video stream is presented. Continuous tracking of targets and handling efficiently occlusions due to objects presented in the scene are considered in this proposed approach. The state of art of the proposed approach can be summarized as:

1. Determination of the interest region of the target in current image.
2. Calculation the color probability distribution of 2D region centered at the search window location in the region of interest (ROI).
3. Back-projection of this histogram with the image in order to obtain the probability distribution image.
4. Application of the Mean Shift algorithm on this image to determine the new target centre in image.
5. Applying an area based estimator.
6. Applying edge detection techniques around the region of interest.

These steps will be explained in the following sub-sections.

Interest region of the target

In the majority of works proposed on tracking methods in general and on the CAMShift method in particular, the definition of the object to be tracked is manually carried out by the operator thanks to a rectangle or an ellipse around the target in image I_t. A convex mask is then used to reduce the spectral influence of the pixels far from the centre which are less reliable than those located near the centre.

In this approach the area corresponding to the target is improved by edge detection techniques for a better object modeling. Edge detection technique [12] in the selected rectangle is performed. The background noise included in the object modeling can be eliminated in order to achieve a better model of the target.

Color probability distribution

The search window is defined around the target and is larger than the target window, increased by a distance d generally ranging between 10 and 20 pixels. The raw image is converted to a color probability distribution image via a color histogram model (ratio histogram) of the color being tracked. The ratio histogram R describes the relationship between the histogram M of the selected region (target histogram) and the histogram I of the whole image or $R = M/I$.

Back-projection

The back-projection is described as a filtering, which leads to the formation of a new grey scale image. The pixel value in the grayscale image is the probability of the ratio histogram defined uniquely by the corresponding chromaticity values of the pixel in the original image. Those pixels whose chromaticity coordinates do not have support from the ratio histogram are attached with value zero. The new position of the object corresponding to the probability distributed image thanks to the Mean Shift algorithm is calculated.

Application of the Mean Shift algorithm

The search window is initially centered at the position of the object in image. The mass centre of the distribution is calculated. If this one is different from the window centre, then the search window moves towards this mass centre. The operation is repeated until the mass centre of the distribution in the window is identical to the window centre. The procedure is stopped if we obtain a minimal variation between the new window position and the preceding one; we can also specify a maximum number of iterations. Dimensions and attitude of the target could be calculated, permitting to adjust its size in case of appearance modifications. But this advantage becomes a problem when a near region of background presents appearance similarities. So a fixed-size target is used, and a fixed-size search window too. Then zeroth moment and the first moment in order to get mass center of search window is calculated as follows:

$$M_{00} = \sum_x \sum_y I(x,y) \quad , \quad M_{00} = \sum_x \sum_y I(x,y) \quad , \quad M_{00} = \sum_x \sum_y I(x,y)$$

Where $I(x,y)$ is the pixel (probability) value in the position (x,y) in the image, and x and y range over the search window. The mass center of search window is (x_c , y_c)

$$x_c = \frac{M_{10}}{M_{00}} \qquad y_c = \frac{M_{01}}{M_{00}}$$

If center of mass is different from the window centre, then the search window moves towards the mass centre. The operation is repeated until the mass centre of the distribution in the window is identical to the window centre. The procedure is stopped if we obtain a minimal variation between the new window position and the preceding one; we can also specify a maximum number of iterations. Dimensions and attitude of the target could be calculated to adjust its size in case of appearance modifications using secondary moment as follows:

$$M_{20} = \sum_x \sum_y x^2 I(x,y) \text{ , } M_{02} = \sum_x \sum_y y^2 I(x,y) \text{ , } M_{11} = \sum_x \sum_y x\, y\, I(x,y)$$

Let

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad , b = \frac{M_{11}}{M_{00}} - x_c y_c \quad , c = \frac{M_{02}}{M_{00}} - y_c^2$$

So the long axis l , short-axis w of tracking objective

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

An area based Estimator

Estimator is used to overcome the sudden change of illumination variation problem as shown in Fig. 1. The size of the search window is controlled by the proposed estimator. The estimator design is based on the fact that "the target size can't be increased to double size between two successive frames". Estimator checks the target centre every frame with the next frame, calculates the error between the two positions; which doesn't exceed 10 pixels in both direction. If it exceeds this value it keeps the target window size as the previous value.

(a) (b) (c)

Fig. 1 *Estimator behavior into tracking algorithm, (a)) frame No. 1405 through a plane flying in cloudy weather, (b) frame No. 1405 resultant from applying CAMShift algorithm, (c) frame No. 1405 resultant from applying AAW tracking algorithm using an estimator to keep tracking although the sudden change of background.*

Edge detection techniques

For creating a tracking window size approximately adaptive to the target size variation along its flight trajectory motion, edge detection techniques are applied within the window created using AAW tracking algorithm as shown in Fig. 2.

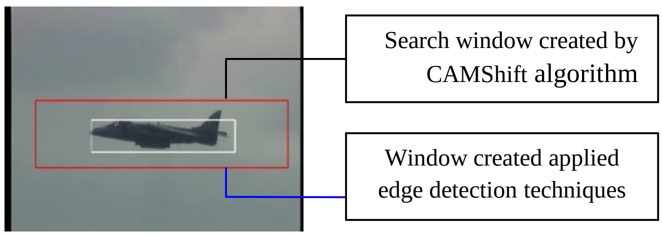


Fig. 2 *Adaptive size tracking window matched the target size using a modified tracking with ED techniques.*

The used edge detector types are:

1. Gradient operators represented on Sobel edge detector.
2. Laplacian of Gaussian
3. Gradient of Gaussian (Canny)

The details of edge techniques will be discussed in the following sections.

Sobel edge detector

The Sobel kernels (named after Irwin Sobel) rely on central differences, but give greater weight to the central pixels when averaging. The computation of the partial derivation in gradient may be approximated in digital images by using the Sobel operators which are shown in the masks below:

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	1

Fig.3 *The Sobel masks*

These two masks together with any of the equations:

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}$$

$$|\nabla f| = |G_x| + |G_y|$$

Are used to obtain the gradient magnitude of the image from the original image.

Canny edge detector

Canny technique is very important method to find edges by isolating noise from the image before find edges of image, without affecting the features of the edges in the image and then applying the tendency to find the edges and the critical value for threshold. I focuses on a particular one developed by John F. Canny (JFC) in 1986 [13]. The algorithmic steps for canny edge detection technique are follows:

1. Smoothing: Blurring of the image to remove noise.
2. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.
3. Non-maximum suppression: Only local maxima should be marked as edges.
4. Double thresholding: Potential edges are determined by thresholding.
5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

Laplacian edge detector

The Laplacian of an image $f(x, y)$ is a second order derivative defined as:

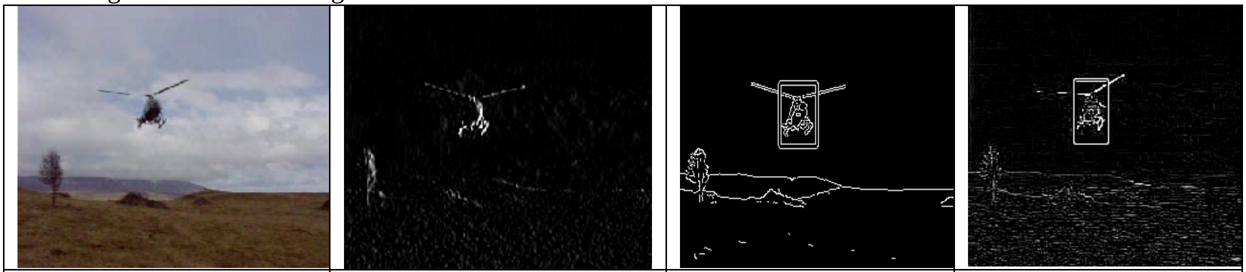
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The digital implementation of the Laplacian function is made through the mask below:

0	-1	0
-1	4	-1
0	-1	0

Fig.4 *The Laplacian mask*

The Laplacian is usually used to establish whether a pixel is on the dark or light side of an edge.
Figure 5 shows the different techniques of applying edge detection to increase the contrast between the edges and the background so that the edge becomes more visible.



(a) (b) (c) (d)

Fig. 6 *Edge detection techniques applied into video sequence; (a) original image, (b) Sobel edge detection applied to image, (c) Canny edge detection applied to image, (b) laplacian edge detection applied to image.*

V. BLOCK DIAGRAM OF MODIFIED APPROACH

As shown in figure 6, the mouse is used to select a rectangle a round the desired target after changing the color format to HSV model. A color histogram is created only once, at the start of the tracking into the selected rectangle concerned with the moving target. This color histogram is used to assign the target probability value to each image pixel in the video frames that follow. As a new video frames arrive, the hue value of each pixel is determined. From that, the target color histogram is used to assign a target probability to the pixel. This process is called histogram backprojection. The target center of gravity is calculated and the rectangle is shifted so it's right over the center of gravity using mean shift algorithm. The secondary moment is used to calculate the size of the target to detect the size of window in the next frame. The tracking window created effected by the saturation and variance parameters, so window differencing techniques is applied to create a precise size window adapted with the target size.

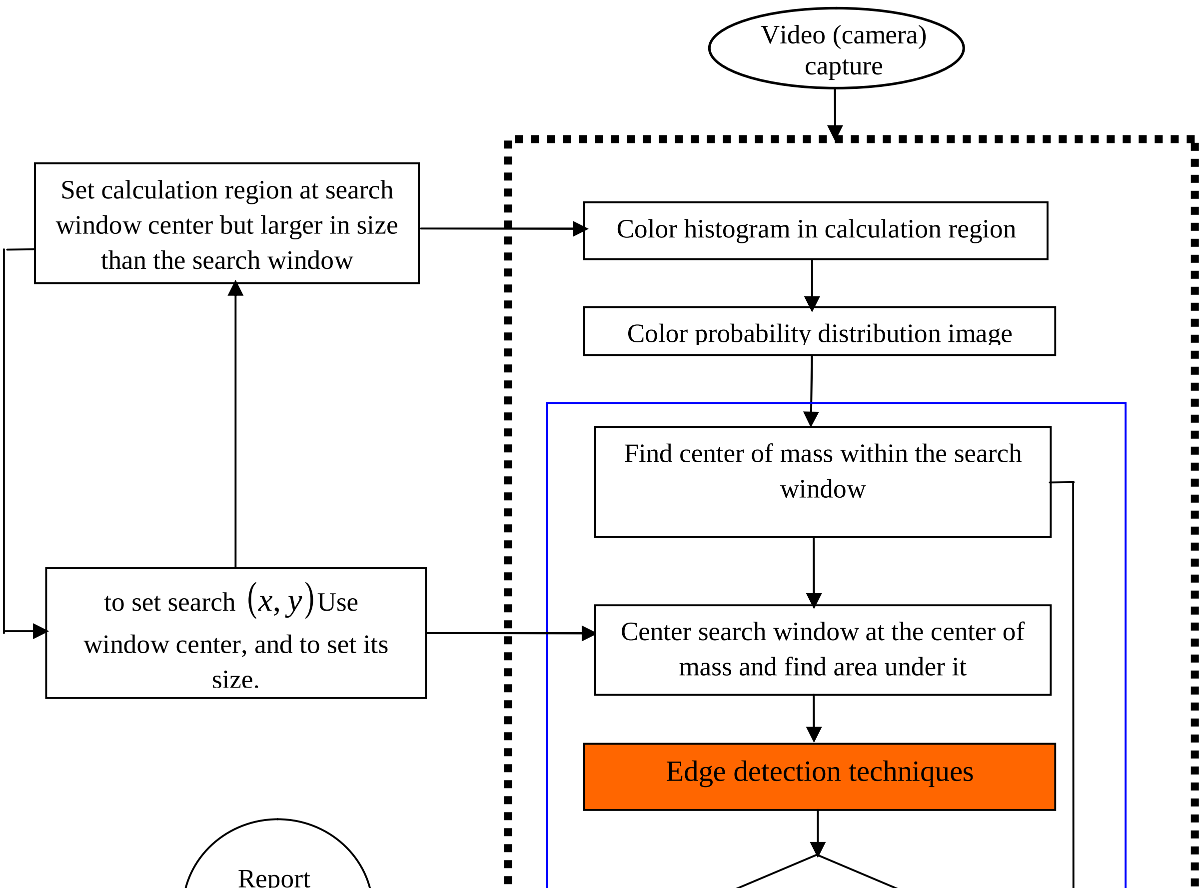


Fig. 6 *Functional block diagram of tracking system*

VI. EXPERMENTAL WORK

Experiments were conducted in real video stream sequences. The program is running on a PC with core 2 Duo CPU, 2.2 GHz and 3 GB memory. In order to obtain real-time performance, the algorithms were developed in OpenCv library interfaces with visual C++ and optimized for faster processing.

Fig. 7 shows an example of tracking a plane video stream (25 frame/second). The results show the efficient handling of the proposed tracking algorithm. The plane is tracked even sudden change of illumination variation occurred.

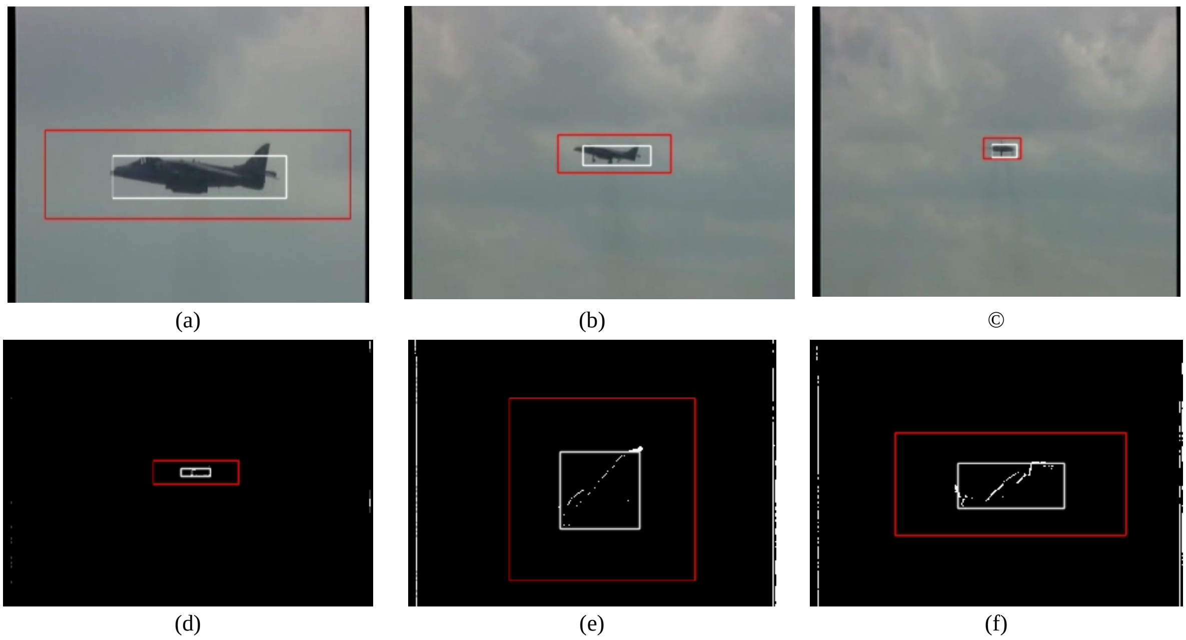
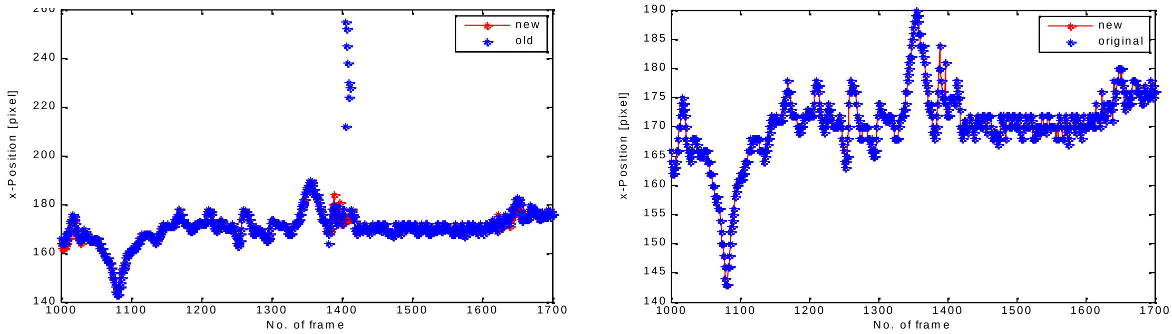


Fig. 7 *Target tracking result for a sequence; (a) frame No. 50, (b) frame No. 350, (c) frame No. 500 (d) back-projection of the frame No. 600, (e) back-projection of the frame No. 850, (f) back-projection of the frame No. 1025.*

Fig. 8 shows the previous experiment tracking trajectory. Fig. 8(a) represents the estimator effect. Without using the estimator, it can be noticed that at the instance of illumination variation due to sudden change in the background color, the tracking algorithm fails to track the target for nearly 14 frames. Fig. 8(b) represents the comparison between the real target center positions with our tracking center. It can be noticed that the robustness of our tracking algorithm that doesn't exceed 2 pixels with the true measurement.

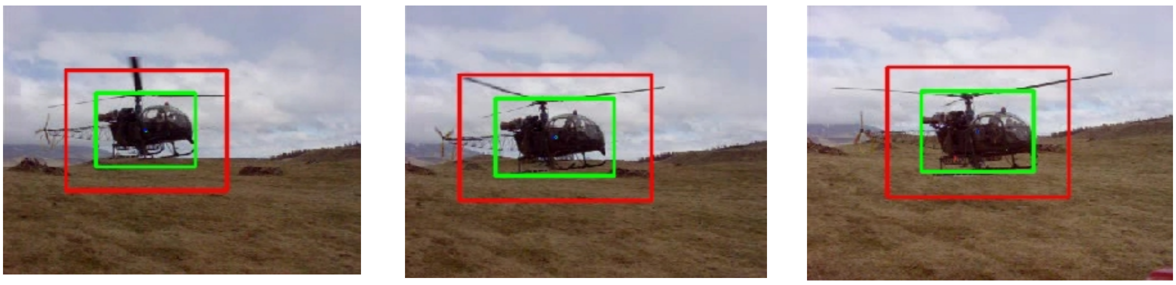


(a) (b)

Fig. 8 *Tracking trajectory of the tracking algorithm, (a) the behavior of applying an estimator (red) and the without (blue), (b) the comparison trajectory of tracking target center (red) with true measurement (blue).*

Figure 9 shows the same pervious example of a plane video stream with 320 x 240 pixels resolution and frame rate 25 frames per second. The results show the efficient handling of the proposed tracking algorithm. The plane is tracked successfully with the inner tracking window resultant from applying canny edge detection techniques that is nearly the same size of the target

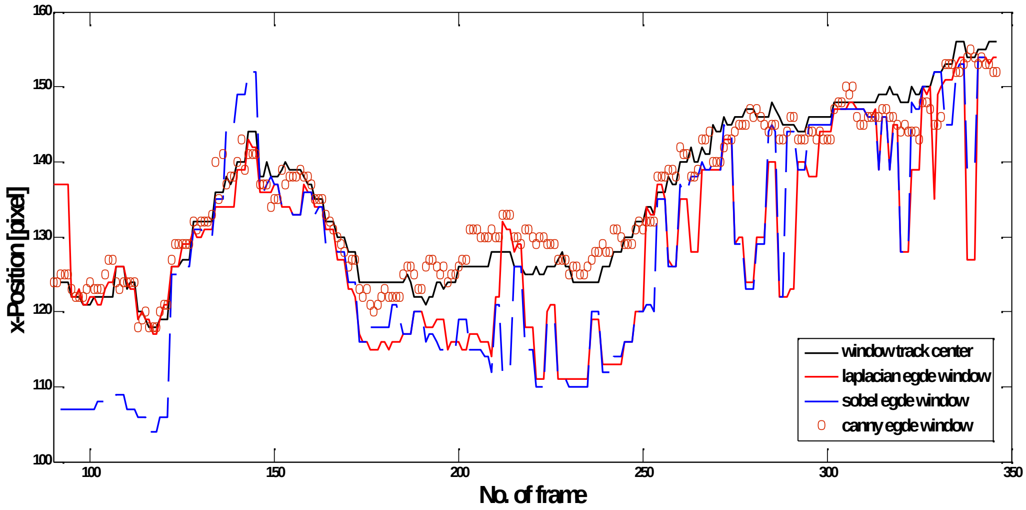
(a) (b) ©



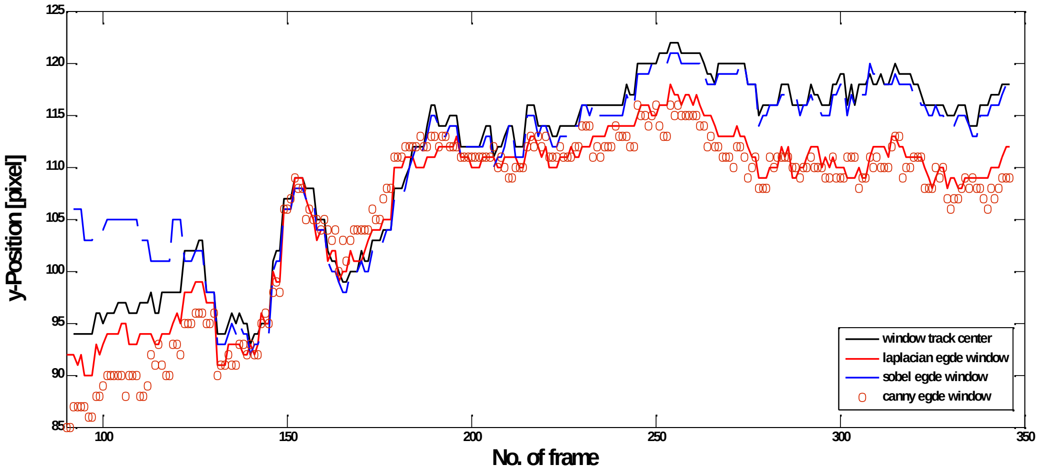
(d) (e) (f)

Fig.9 *Target tracking result for a sequence; (a) frame No. 25 shows canny edge detection technique on the image, (b) frame No. 100, (c) frame No. 150 (d) frame No. 200, (e) frame No. 250, (f) frame No. 325.*

Figure 10 shows comparison target center of gravity trajectory between the outer window tracking and the inner window created by different edge detection techniques in both direction x and y. it's shown from result that canny edge detector is more accurate more than sobel edge and laplacian edge. According to applying canny edge the center of gravity in both directions doesn't exceed 5 pixels in both directions.



(a)



(b)

Fig. 10 Motion trajectory comparison between outer and inner track window, *(a) the target center in x-direction, (b) the target center in y-direction.*

Table 1 shows the CPU clock time speed of the execution process for different edge detectors technique.

Edge type	**Sobel edge detection**	**Canny edge detection**	**Laplacian edge detection**
Delay time per frame	**1.51** msec	3.22 msec	5.73 msec
Frame rate	**24** frame /sec	23 frame /sec	22 frame /sec

Table 2 *CPU clock time speed of the edge detector technique*

From figure 10 and table 1 we can able to choose the best technique for our application. For real time application sobel edge detection technique is the best one hence it has the lowest Delay time per frame. For other application that doesn't in need to reality; canny edge detection technique became the most common used due to its nearest behavior to the true case.

VII. Conclusion

In this paper, target tracking algorithm using adaptive window technique is introduced. In the proposed approach the CAMShift algorithm is modified to be used for tracking targets. A novel and efficient approach for dealing precisely with tracking window size is presented. The effectiveness of this approach is also appeared in dealing with sudden change due to illumination variations, and the manual adjustment problems of the environmental target parameters. Future work includes increasing the tracking performance of the system by developing multi-target tracking in a single board computer.

REFERENCES

[1] G. R. Bradski. Computer vision faces tracking for use in a perceptual user interface, 1998.

[2] Y. Cheng. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell., 17(8):790–799, 1995.

[3] A. S. K. Bin and L. Y. Kang. Face detection and tracking utilizing enhanced camshift model. International Journal of Innovative Computing, Information and Control, 2006.

[4] J. G. Allen, R. Y. D. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In VIP ’05: Proceedings of the Pan-Sydney area workshop on Visual information processing, pages 3–7, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[5] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In CVPR, pages 750–. IEEE Computer Society, 1997.

[6] J. Campbell. Shaogang gong, stephen j. mckenna, Alexandra psarrou, dynamic vision: From images to face recognition - review. Artif. Intell. Rev., 14(6):619–621, 2000.

[7] R. Stolkin, I. Florescu, and G. Kamberov. An adaptive background model for camshift tracking with a moving camera. In Proc. of the 6th International Conference on Advances in Pattern Recognition, 2007.

[8] C. Zhang, Y. Qiao, E. Fallon, and C. Xu. An improved camshift algorithm for target tracking in video surveillance. In Proceedings of 9th. IT and T Conference, number 12, 2009.

[9] Robin Hewitt. SERVO Magazine series, seeing with OpenCv. Part 3. March 2007.

[10] Karan Gupta, Anjali V. Kulkarni. Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching. IEEE Trans. Pattern Anal. 2008.

[11] G. John Allen, Y. D. Richard Xu and S. Jin Jesse, “Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces”, Inc. Australian Computer Society, vol.36, 2004.

[12] Daggu Venkateshwar Rao, , Shruti Patil, Naveen Anne Babu and V Muthukumar “Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages”, International Journal of Theoretical and Applied Computer Sciences, Vol.1, 2006.

[13] V. Torre and T. A. Poggio. “On edge detection”. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 2, pp. 187-163, Mar. 1986.