REAL TIME TRACKING IN 3D SPACE BY ROBOTIC VISION

KHALID AL-KHATEEB AND MOHD. AKRAM DANDU

Department of Electrical and Computer Engineering
Kulliyyah of Engineering
International Islamic University Malaysia
Email (khalid@iiu.edu.my, ik@iiu.edu.my)

ABSTRACT

A digital real time auto tracking robotic system is designed and analyzed. The tracking is based on passive detection using live TV imaging. The target coordinates are determined by an image processing technique and non-linear prediction algorithm. The control signals derived from the video signal set the azimuth ? and elevation f of the camera position to track the centroid of the target on the screen. The time development of the aim point in successive images represents the tracking path. Non-linear prediction determines the location of the target in the next TV frame by extrapolation. For distant objects a range finder can be mounted with the camera to determine the coordinate, r. Hence, a set of spherical coordinates (r, ?, f) for the target are determined.

1. INTRODUCTION

Object tracking with passive detection is an important capability in robotic vision systems. These systems have potential applications in many industrial fields. One of the aims of this work is to build tracking system using standard equipment and components so that essential tests can be readily carried out on various proto-types. It should be able to perform satisfactorily in real-time. Tests are necessary for the analysis of the performance under real operational conditions. Real-time processing for the purpose of this work is limited to one or two frame times of the TV images.

Auto tracking of a fast moving maneuvering target presents numerous problems, which require a capability of responding to fast changes within the available time period. Obviously this is more stringent than the requirements of stationary robots, or robots with predetermined movements. Hence, effort is made to minimize the computational load to a unit frame time, by reducing the complexity of the processing required by dynamic images. One approach is to segment the image and perform fine analysis of the segment that surrounds the target, by omitting the rest of the image. Different approaches to this problem [1-6] met with various degrees of success. However, much of the pertinent results are either not published, or not efficient enough to cope with the complex problem of tracking in 3D- space, in real time.

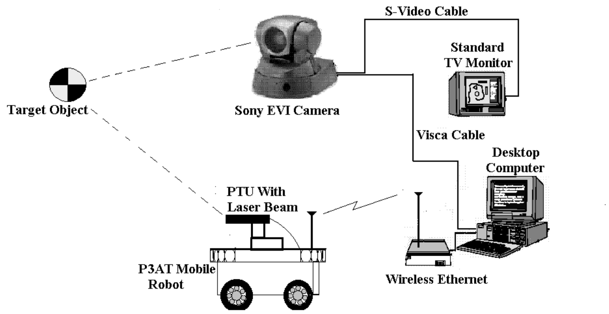


Fig. 1, Real Time Tracking System in 3D-Space

Under realistic conditions, as the presence of occlusions, changes of illumination etc., the problem becomes even more complex.

This kind of processing raises major concerns among researchers in the fields of tracking, surveillance, automated guidance systems, autonomous robot navigation, inspection, and monitoring. Motion detection and tracking are capabilities, which may require robotic vision as well as high-level machine perception and scene understanding [7]. In attention cues, processing can be concentrated on the region of interest. Limiting the computations to the relevant part of the visual field is recommended for systems operating in real time, since image-processing with increased computational time and subsequently longer time delay, may altogether lead to deterioration in real time system performance [8].

Published work related to flow-computation, in most cases, does not mention the computational cost [8-11,13]. This makes proper evaluation and selection of an algorithm suitable for real time implementation rather difficult. Although there are methods for real-time motion detection and tracking [12], mostly the camera movement has been constrained to rotation about one axis only, and as the number of degrees of freedom increases, calibration becomes less reliable.

Detection and segmentation of a sequence of images containing a moving object [1, 2] has been performed with reasonable success using a hybrid technique that combines analog and digital processing. It has also been performed with flow methods [8-11,13]. The computational cost however, is so high that real-time performance is difficult to achieve with standard low cost equipment and components. Other claims [14] of real time tracking gave a vague presentation of experimental and performance figures, especially in relation to frame-rate, which is essential in evaluating the overall performance. An optimal frame rate that minimizes the computational cost per unit time can be calculated [3], to maintain the object image under track with a certain probability [8,14] while keeping the computational capacity requirement to a minimum.

2. CENTROID TRACKING ALGORITHM

Moving Target Detection (MTD) and acquisition algorithm aims at extracting the image of the moving object from the rest of the scene. When the movement is independent from the background, the background can be removed by a subtraction technique.

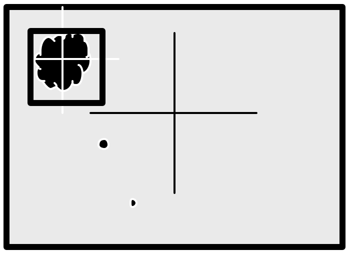


Fig.2, Determination of Target Coordinates

Threshold segmentation is a relatively easy method to generate the appropriate image for the object of interest. The least intensity value N(x), the maximum intensity value M(x) and the largest mean inter-frame difference value D(x) for each pixel in the over all frame, collected in the previous step are stored as the background scene information. Separation in each iteration of background subtraction, represents a pixel x from the image I where the modeled variance k is empirically determined. All other pixels are classified as background.

$$\left|I^t(x)-M(x)\right| \geq kD(x) \vee \left|I^t(x)-N(x)\right| \geq kD(x)$$

…………………………………. (1)

The centroid of this region is taken as the position of the aim point for the target in the image plane. Centroid tracking is useful for tracking bounded objects, which can be fully contained within a tracking window. The centroid of the target is determined by computing the geometric center of the target in the image plane, Fig.3. The tracker integrates horizontally and vertically, and the coordinates are defined by the point of intersection of the two datum lines parallel to the x-axis and y-axis respectively that divide the area into two equal halves. The digitized data of the video signal is used in the computation of the centroid within the tracking gate.

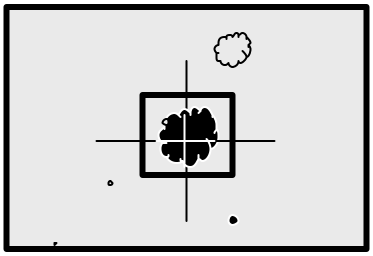


Fig. 3, Electronic window illustrates the algorithm

In this algorithm all the edges of the target are defined. The coordinates of the center or the tracking window are:

$$\hat{x}_t = (\hat{x}_{Rt} - \hat{x}_{Lt})/2$$
$$\hat{y}_t = (\hat{y}_{Bt} - \hat{y}_{Tt})/2$$

………………………… (2)

These represent the average between; Right and Left edges and Bottom and Top edge of the target respectively.

This technique is equally viable for a white object against a dark background as for a dark object against a bright background, so that the integration within the window computes the center of mass for the image, which is the true centroid of a two-dimensional object. The video frame is broken into a matrix by utilizing (X,Y) television scan signals and a high speed clock. The information obtained by the centroid tracker is also used in the Target acquisition mode to set the initial position of the tracking window and its size in accordance with the target position and size. The auto-tracking mode utilizes the centroid information even when the target is fragmented. Centroid tracking is suitable for most applications, and can have wide use in various surveillance systems. The target region ideally consists of all the pixels of the object image. The background region contains pixels in the background immediately surrounding the target. Based on the statistics of these two regions, a decision rule is formed so that each image pixel is classified as either a target or background. A segmenting threshold level is computed as the average of the mean values of the background and target regions. The result is a binary image in which target pixels are represented by 1, and background pixels by 0. The centroid of the target is computed within the window of the target region from the binary image. The object centroid c (x_t , y_t) is then used as the aim point.

$$c(i,\ j) = ColorMatch(R(i,\ j), G(i,\ j), B(i,\ j)) \quad \ldots$$

............................(3)

$$\hat{x}_t = \frac{\sum\sum_{(i,j)\in TG} i\ .\ c(i,\ j)}{\sum\sum_{(i,j)\in TG} c(i,\ j)}$$

$$\hat{y}_t = \frac{\sum\sum_{(i,j)\in TG} j\ .\ c(i,\ j)}{\sum\sum_{(i,j)\in TG} c(i,\ j)}$$

.... (4)

where,

$\hat{x}_t$	Target centroid estimate in the x direction
$\hat{y}_t$	Target centroid estimate in the y direction
$i,\ j$	Image coordinates in the x and y directions
$c(i,\ j)$	Pixel classifying function

0 if pixel (i, j) is background
1 if pixel (i, j) is object

*TG is a s*et containing all pixels in the target region.

If the segmentation process is accurate, most, if not all of the target and background pixels will be correctly classified. In such a case the estimated centroid will be very close to the actual position of the target.

3. ROBOT CONTROL SYSTEM

In order to lock onto the moving target, the algorithm provides a drive signal to the pan-tilt of the camera, so that as the target moves in the object plane, the camera will move in such a way that a corresponding movement of the target in the image plane will keep it at the center of the screen of the monitor. In the general case, the angle of vision of the camera is quite small. Hence, there relation between the angle of vision in the object plane and the displacements in the image plane respectively will be linear, i.e. ? = tan ? = sin ?, and f = tanf = sin f . The algorithm is based on the construction of a 2-D look up table (LUT) that gives the values of pan-tilt angles in degrees, which correspond to the number of pixel displacements of the image in the x and y directions, as illustrated in Fig 4.

The image-based approach shown in Fig. 4 and Fig. 5, with the 2-D LUT is simple, accurate, fast and easy to implement, where the algorithm is basically the mapping between the pan-tilt angles and the pixel displacements. Therefore it only requires a reading of the number of pixels, which is purely image-based, and does not impose many restrictions on the specifications of the camera.

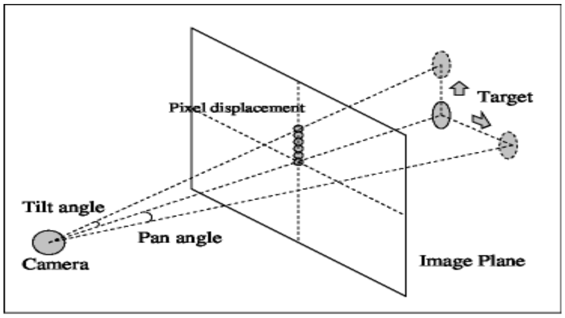


Fig. 4 shows the correspondence between the object plane and the image plane.

The components of the LUT are determined irrespective of the distance between the camera and the object.

One of the ways to determine the range of the target is simply by calibration of the zoom signal [2] for a certain size object, or by adding a range finder.

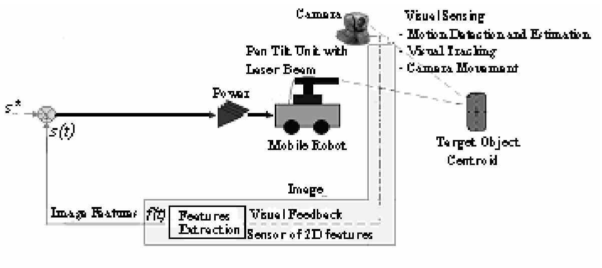


Fig. 5, Robot Control System

The operation of the present system (Fig. 1) has been tested in the laboratory using a laser beam mounted on a mobile robot. The laser beam is controlled by another pan-tilt mechanism. It points to the target automatically when lock-on is accomplished. The host computer computes new values for the coordinates of the target, assuming a transformation of origin from the position of the camera to the position of the laser. Then the signals for the azimuth and elevation angles of the laser beam ***(?₁,f ₁)*** are communicated to the robot via wireless link.

The motion vector in the image plane (Fig.2) is defined by the displacement between the center of the screen and the centroid of the target image. The geometry is shown in Fig. 6:

$$T_1T_2 = (\hat{x}_t, \hat{y}_t) - (W/2, H/2)$$
……………………………… (5)

where W and H are the width and height of the TV frame respectively.

Accordingly, the signals for the new position of the target will be:

$$T_1T_2(pan) = (\hat{x}_t - W/2, 0)$$
$$T_1T_2(tilt) = (0, \hat{y}_t - H/2)$$
……………………………… (6)

As the object moves, the centroid of the image is continually displaced from the center of the image plane. It is followed by the tracking algorithm that turns the camera head, which is driven by appropriate control signals to the pan-tilt motors.

Initial calibration begins by assuming that the object is in the (X,Y) plane and the camera is at coordinate (0,0), i.e. pan-tilt angles are zero degrees. The normal distance from the camera to the (X,Y) plane will vary as the target moves from T1 to T2. The coordinates of the target hence become (X,Y,Z). The coordinates of the PTU of the laser mounted on the robot R are *(x,y,z)* located at a fixed distance h from the camera C, which is at the origin (0,0,0) of the coordinate system.

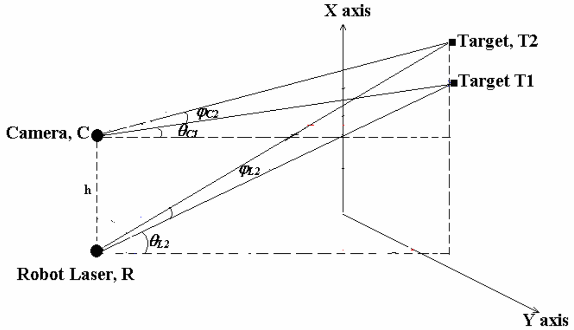


Fig. 6 shows the relative positions of the Target T in the object plane, the Camera C and, and the Robot R.

If the centroid of the object image remains at the center of the image plane, it ensures the target to be always within the field of view of the camera. Hence, the laser beam can always be directed towards the target, as explained below.

A point P in 3-D space can be defined by *(r, ? , f)* in spherical coordinates and by *(x,y,z)* in rectangular coordinates. The relation between them is:

$$\begin{cases} x = r \sin q \cos f \\ y = r \sin q \sin f \\ z = r \cos q \end{cases} \quad \text{……………………………… } \mathbf{(7)}$$

and

$$\begin{cases} r = \sqrt{x^2 + y^2 + z^2} \\ f = \tan^{-1}(y/x) \\ q = \cos^{-1}\left(z / \sqrt{x^2 + y^2 + z^2}\right) \end{cases} \quad \text{……………………… } \mathbf{(8)}$$

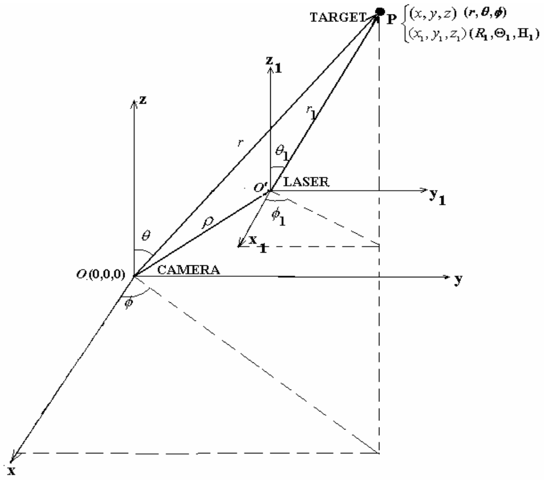


Fig. 7, Target tracking and estimation

Therefore, the laser beam is directed towards the target T, by calculating *(r, ?, f)* in a coordinate system whose origin has been shifted from (0,0,0) to R at (x,y,z) corresponding to $(r_1, ?_1, f_1)$. Actually, only the new values $(?_1, f_1)$ for the target will be sufficient.

Generally, the origin of the frame of reference remains centered at the camera, but the robot can change its position and maintains the capability of continuously pointing the laser at the target. Although some robot laser control systems translate and rotate the image in the TV frame to track the target, the added complexity may reduce the reliability of the system.

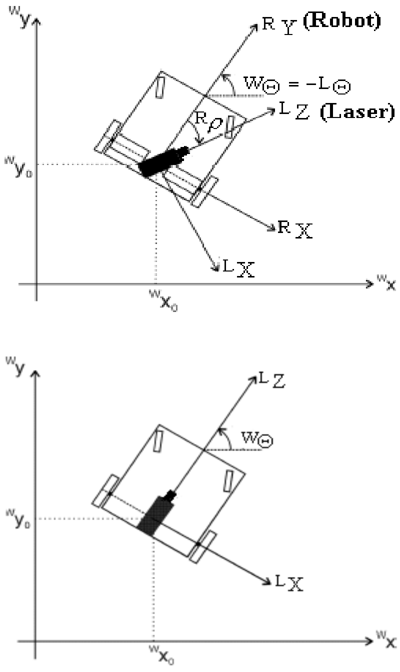


Fig. 8, The direction of the robot is controlled by a local signal from the pan-tilt of the laser

4. PRACTICAL IMPLEMENTATION

Reliability is a major issue in tracking systems, which becomes essential in real time applications. To achieve reliable tracking in real time based on this imperative condition, this study adopted a solution of hardware and software implemented together. The location of the target in successive TV frames, itself represents the trajectory of the target, which lead to the development of suitable algorithms. This section describes the construction of the system hardware.

Operation of the system is represented by the flow chart in Fig. 9. Initially, an image is selected for the construction of the LUT. Any single-colored object that has sufficient color-contrast with the background in the scene can be used for this purpose. On initialization, pixel classification and location of the target are computed and stored. After the target is initialized, an estimate of target location is computed using target dynamics. Then, the tracking is achieved using pixel classification-based algorithm. It gives the coordinates of the centroid. The next step is the movement of the camera, so that the centroid appears at the center of the image plane, and a reading of the pan-tilt angle is recorded. The tracking algorithm moves the PTU incrementally by one or more pixel at a time, depending on the required accuracy, and reads the angle each time the center of mass changes position in each direction. The difference between the current angle and the stored value is also stored at a corresponding bin in the LUT. This step is iterated until the pan-tilt angles for all possible points are acquired.

5. EXPERIMENTAL RESULTS

Testing of the tracking algorithm was done using a TV camera connected to the host PC and TV monitor. The host PC is serial RS232 cable connected to the pan-tilt unit of a laser mounted on a mobile robot. Sony EVI-D31 and pan-tilt unit (PTU D46-17) are used for object tracking. Two types experiments were conducted; the first demonstrates the capability of the tracking system and the second demonstrates the accuracy of the control system. The plot of target speed setting Vs speed (deg/sec) is shown in Fig. 10. It corresponds with Sony EVI camera technical specifications.

Acceptable performance is achieved with standard equipment. It takes 33 ms to grab a frame, display it on the screen, process the image, identifies the object and moves the camera. The tracking method proved to be reliable, with reasonable accuracy. The defined coordinates of the mobile robot with respect to the TV camera is important for the system to determine the distance from the robot to a target. As the target moves the video system was able to locate it dynamically, and the pan-tilt control system of th le to illuminate it at each location.

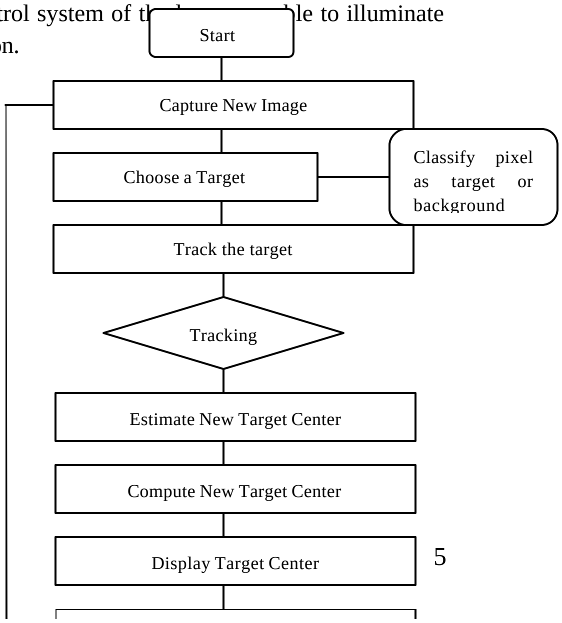


Fig. 9 Flow chart of operation

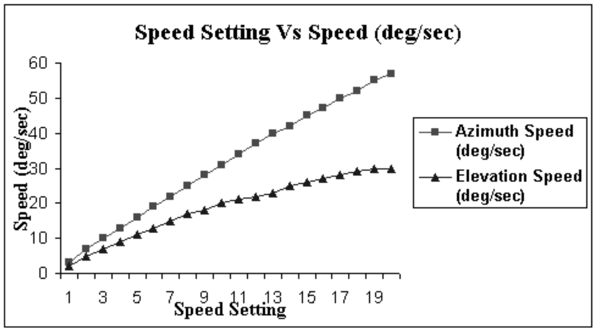
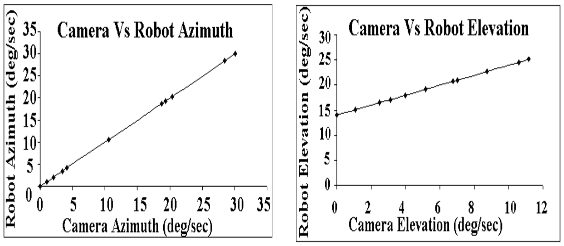


Fig. 10, Azimuth and elevation speeds



(a) (b)

Fig. 11, Comparison between camera and robot angles (a) azimuth, (b) elevation.

7. CONCLUSION

The design, implementation and testing of a real time tracking system capable of tracking maneuvering objects in 3-D space has been achieved using standard of the shelf components. The system is based on the concept of centroid tracking, which is fast enough for operation in real time without compromising the accuracy. Background subtraction, and adaptive thresholding, which reduce the computation time and facilitate the calculation of the centroid coordinates, are the main features of the system.

To ascertain proper performance of the system, the laser pointer mounted on a mobile robot was able to indicate the exact location of the target in a 3-D coordinate system.

ACKNOWLEDGEMENT

The authors wish to thank the International Islamic University Malaysia Research Centre at Gombak, Malaysia, for funding this research work.

REFERENCES

[1] K. A. Al-Khateeb, F. Al-Naima, and A. N. Mardan "Adaptive Real time TV tracking System with Prediction Capability", ROVPIA '94, Ipoh, Malaysia, 1994.

[2] K. A. Al-Khateeb, "Improved Sensitivity of Real-Time Processor for Tracking by TV Imaging", Second Jordanian Conference on Electrical Engineering Mu'tah, Jordan, 1996.

[3] S.M. Hong "Steady State Analysis of Computational Load in Correlation-Based Image Tracking", IEE Proc.Vis. Image Signal Process. Vol. 149, No. 3, 2002.

[4] T. Gevers, "Robust Segmentation and Tracking of Colored Objects in Video". Circuits and Systems for Video Technology, IEEE Transactions, pp 776-781, June 2004.

[5] T. He, P. A. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic and T. Abdelzaher. "Achieving Real-Time Target Tracking Using Wireless Sensor Networks", the 12th IEEE Real-Time and Embedded Technology and Applications Symposium, Apr. 2006.

[6] V. Kravtchenko and J. Little "Efficient Color Object Segmentation Using Diachromatic Reflection Model" Proceedings of the IEEE PACRIM '99, pp 90-94, 1999

[7] J.L Crowley, and H.I. Christensen, "*Vision as Process-Basic Research Series"* Springer Verlag, New York, 1995.

[8] P. Nordlund, Thomas Uhlin. "Closing the loop: Detection and Pursuit of a Moving Object by a Moving Observer" Computational Vision and Active Perception Laboratory (CVAP), pp. 2-6, 1995.

[9] J. L Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of Optical Flow Techniques", International Journal of Computer Vision, pp. 43-77, 1994.

[10] M. Irani, and S. Peleg "Motion Analysis for Image Enhancement: Resolution Occlusion, and Transparency" Journal of Visual Communication and Image Representation, pp. 324-335, 1994.

[11] J. Y. A Wang,.and Andelson, Edward H. "Spatio Temporal Segmentation of Video Data" Proc. Of the SPIE: Image and Video Processing II, volume 2182, San Jose. 1994.

[12] D. Murry and Basu, "Motion Tracking with an Active Camera", IEEE Trans. Pattern Analysis and Machine Intell.,Vol. 16, No. 5, pp. 449-459, 1994.

[13] Meyer, F.G. and P. Bouthemy, "Region-Based Tracking Using Affine Motion Models in Long Image Sequences" Computer Vision, Graphics, and Image Processing, Vol. 60, no. 2, pp. 119-140, 1994.

[14] Y. Rosenberg and M. Werman, "Real Time Object Tracking from a Moving Video Camera: A Software Approach on PC", Workshop on Applications of Computer Vision, Princeton, pp. 238-239, 1998.