Proc. Intelligent Robotic Systems '94, Grenoble, France, July, 1994

Real-Time Model-Based Visual Tracking

Henry Schneiderman, Albert J. Wavering, Marilyn Nashman, and Ronald Lumia

Building 220, Room B127
Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, Maryland 20899

Abstract

This paper describes a robust approach to real-time model-based visual tracking and servoing for rigid polyhedral objects moving with 3 degrees of freedom. It is recognized that low-level feature segmentation is not infallible. Uncertainty measures enter into all computations to account for varying degrees of confidence in the measured object features. Robust tracking and servoing has been demonstrated in the presence of partial occlusion and cluttered backgrounds at tracking speeds of up to 1.2 rad/s (69 deg/s). This algorithm has also been used for vehicle following where a vehicle is driven autonomously by following the path of another vehicle.

1. Introduction

The visual tracking problem has received considerable attention from the robotics and computer vision communities. There are many variations on the problem depending on the assumptions made about the target. Most work has focussed on tracking rigid objects. However, some attention has been given to the tracking of deformable objects [1]. There are also differences in terms of how much is assumed known about the appearance of the target and the motion of the target. In this work it is assumed that we are tracking a rigid 3-D object with known geometry and appearance and moving with 3 degrees of positional freedom.

Approaches to visual tracking also vary in terms of the assumptions made about the frame to frame motion of the target. Many works, including the one described in this paper, assume spatio-temporal continuity; i.e., the location of the target in incoming images can be approximately predicted using the previous location and the estimated motion of the target. References [2] and [3] [4] are also examples of work that make this assumption. In another method, [5], described as robust to large and arbitrary motions between images, line detection is performed on the raw image. The extracted lines are then compared with the model, and a best-first search algorithm is used to find the best set of one-to-one correspondences between the model lines and the extracted lines. The method described in [6] is also less restrictive in terms of the spatio-temporal continuity requirement. Feature location is determined using a sum-of-squared differences optical flow algorithm. This algorithm searches the entire image for each feature. Unless the motion greatly distorts the tracked features, the algorithm should be able to handle displacements within the field of view. This method also uses a unique control formulation where the target motion is continually estimated as an ARMAX model [7] and the estimated parameters are used to adaptively adjust the gains of the manipulator servo controller (a manipulator is used to point the camera).

Our method emphasizes robust methods for detecting the target. It is recognized that low-level feature segmentation is not infallible. The principles used are motivated by statistics. At the earliest stage of processing (see 2.3), the location of each feature is determined separately. These features are combined — accounting for the variance in each feature measurement — to obtain the best linear unbiased estimate (BLUE) of target location. The estimates of the target location are combined over time — accounting for the covariance in target location — to update motion models for the target. The motion models are then used to predict target position and feature locations in the next incoming image. Prediction of target motion is also necessary for the purposes of control to compensate for the delays incurred by image acquisition and image processing.

The algorithm has been demonstrated for target tracking using the TRICLOPS camera pointing system and for vehicle following. In vehicle following, the position of a lead vehicle is visually tracked from the vantage of a pursuing chase vehicle. This algorithm has achieved very promising performance at tracking targets moving at significant speed and in the presence of partial occlusion, shadows, outdoor lighting conditions, and arbitrary background scenery.

2.0. The model-based tracking algorithm

An overview of the processing scheme is shown schematically in Figure 1. A single CCD camera provides video images to the system at the rate of 30 hertz. Image data flows through the diagram as indicated by the arrows. There are several successive stages of computation:

1. Edge extraction - Extracting position and orientations for all edge points.
2. Data association - Determine likely groupings of edge points to each model feature.
3. Feature measurement - Use grouped edge points to determine location of each feature.
4. Feature aggregation - Determine the overall location of the target by fitting the target model to the conglomerate of computed feature locations.
5. Update motion models - Use computed object location to update parameters (e.g. velocity, acceleration) that describe how the target moves as a function of time.

6a. Predict target location in next image - Extrapolate motion model to predict target location in next image. Use predicted target location to determine corresponding predicted feature locations.

6b. Predict target location at next servo cycle - Predict target location such that trajectory of camera pointing device (TRICLOPS) will keep image of object centered in the field of view.

These processing steps are described in the remainder of this section.

2.1. Geometric model of target

Geometrically, the target is modelled in terms of its edge features i.e., light-to-dark transitions. Each of these edges is modelled by a line segment. One such model is shown in Figure 2 for the target shown in Figure 2.

2.2. Initialization of tracking

Registration between the model and the target is initially established by a teleoperator. Using a graphic representation of

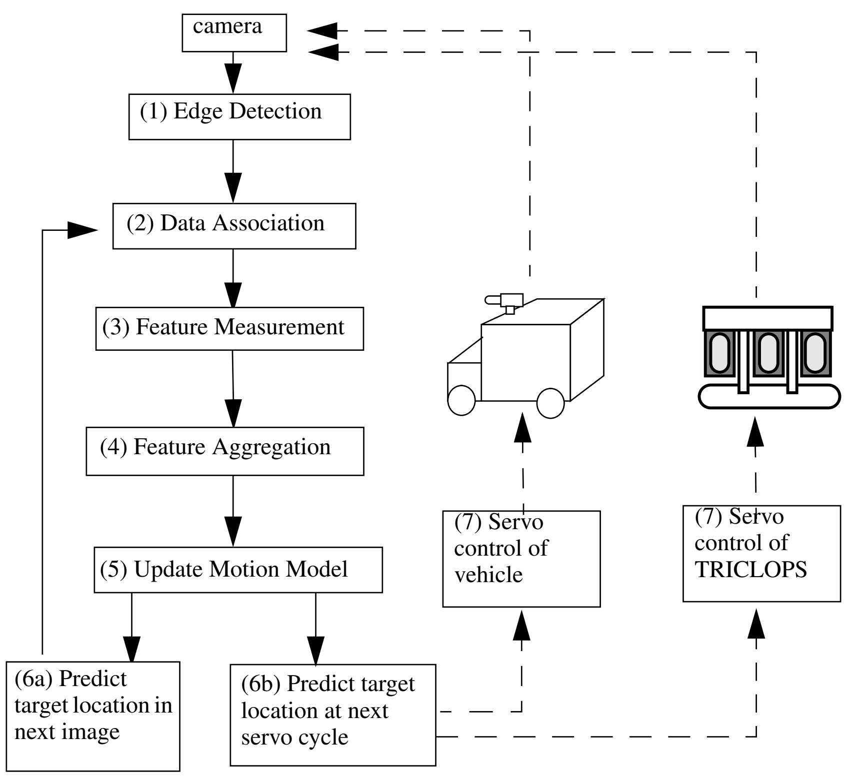

Figure 1. Processing Overview

the model superimposed on the live video image, the teleoperator positions and scales the model such that it graphically aligns with the image of the target.

2.3. Edge detection

In the first processing step, edge detection, is performed on the image. The image is separately convolved with the two 3x3 Sobel gradient operators. The resulting gradient images are used to compute edge magnitude and edge direction for every pixel. The magnitude image is then thresholded to produce a binary edge image.

Since it can be shown that a statistical estimate will improve — the variance in the estimate will decrease — when more data are considered [9][10][11], the resulting binary edge image is subsampled as little as possible so that the subsequent estimates of feature location (2.4) are as good as possible. Currently, the edges within the window of interest (see 2.4) are randomly sampled such that a maximum not exceeding 1500 points are acquired. The upper limit of 1500 points is necessary to ensure that the timing constraints of the system are satisfied.

2.4. Data Association

Many of the detected edge pixels are not caused by model features. There are many entities in a scene that can give rise to edges. In outdoor scenes in which vehicle following was performed, strong edges were produced by telephone poles, shadows and power lines. For tracking experiments in the laboratory, strong edges were caused by bookshelves and tables.

The purpose of this processing is to best determine which edge pixels match each model line segment. In order for an edge pixel to be grouped to a model feature, it must satisfy two criteria. The first criterion is the two-dimensional spatial proximity of the edge point to the model feature. The second criterion is the similarity of direction of the edge point with the angular orientation of the model feature. An edge pixel is discarded if it does not satisfy both criteria for any model features. The predictions of model feature locations are computed during the previous processing cycle from the prediction of the overall target location, (6a) in Figure 1.

In each image, many edges can be discarded immediately on the basis of the spatial proximity criterion. This is done by eliminating all edges that fall outside a window of interest. This eliminates many, but not all edges that violate the spatial proximity criterion. For all edge points falling within the window of interest, the two data association criteria are applied on a point by point basis.

2.5. Feature Measurement

After edge pixels have been grouped to each model feature, the location of each feature is determined such that it gives the best fit to its group of edge pixels.

The location of each model feature is represented by two parameters (*m, b*). One of two representations is chosen, depending on the predicted orientation of the feature:

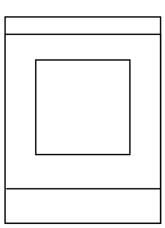

Figure 2. Model of target

Figure 3. Target

$$
\begin{aligned}
y &= mx + b \qquad \text{horizontal representation} \\
x &= my + b \qquad \text{vertical representation}
\end{aligned} \tag{1}
$$

If the horizontal representation is used, *(m, b)* are determined by minimizing the least squares residual in the following set of overdetermined linear equations:

$$
\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} \tag{2}
$$

$$
b \quad = \quad A \quad x
$$

(x_i, y_i) are the coordinates of the i^{th} edge point.
n is the total number of edge points.

For a vertical representation, the x and y coordinates of the edge points are interchanged.

There are many least squares methods that exist for solving this set of equations. One such solution is expressed by:

$$
\hat{x} = (A^T A)^{-1} A^T b \tag{3}
$$

2.6. Feature Confidence

Some computed feature locations will be more accurate and reliable than others. The differences in reliability are consequences of the fallibility of the data association algorithm. Since edge pixels are only discriminated on the basis of spatial and angular proximity, it is unlikely that all spurious edges will be discarded.

It is valuable to know how much trust can be placed in each of these computed feature locations. One way this can be measured is in the empirical variance in feature location as given by:

$$
\sigma_{\hat{x}}^2 = \frac{1}{n}(b - A\hat{x})^T (b - A\hat{x}) \tag{4}
$$

A large variance corresponds to a scattered distribution of edge points about the computed feature location. Such a scattering of edge points may indicate the presence of spurious edge points. This will happen when edge pixels from two visual entities (e.g. the desired target edge and an undesired background edge) are grouped to a model feature. Figure 4 shows a typical scene of target tracking with a bookshelf in the background. Figure 5 shows a binary edge image (edges indicated by black) of the target when it is in front of the bookshelf. Near the left-most edge of the model, there is another strong edge caused by the background. It is likely that both sets of edge pixels will be grouped to this left-most line segment in the model. If this happens, the variance in this line segment's location will be higher than for other line segments. This feature will then carry less weight when the features are aggregated to determine the overall target location (see 2.8).

2.7. Corner Features

The location of all corner features are computed by the intersection of the appropriate pairs of line features computed in 2.5. For example, for two lines represented by the horizontal representation, the point of intersection $v = (v_x, v_y)$ is computed by solving the set of equations for v:

$$
\begin{bmatrix} -b_1 \\ -b_2 \end{bmatrix} = \begin{bmatrix} m_1 & -1 \\ m_2 & -1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{5}
$$

$$
w \quad = \quad F \quad \mathrm{v}
$$

Since corner position is a linear function of the line parameters, the covariance in v is then given by [9][10]:

$$C_v = F^{-1}\Sigma F^{-T} \qquad\qquad \Sigma = \begin{bmatrix} \sigma_{\hat{x}1}^2 & 0 \\ 0 & \sigma_{\hat{x}2}^2 \end{bmatrix} \tag{6}$$

2.8. Feature Aggregation

In this stage of processing, the overall target location is computed that gives the best fit of the target model to the conglomerate of computed corner feature locations. The target was assumed to be moving with only 3 degrees of positional freedom with respect to the camera (orientation of the target is assumed fixed with respect to the camera). More specifically, the target location is determined by finding (c_x, c_y, $1/c_z$) such that this weighted least squares residual is minimized:

$$J_l = \sum_{i=1}^{n} \frac{1}{C_{vi}(1,1)}\left(c_x + \frac{1}{c_z}p_x u_{xi} - v_{xi}\right)^2 + \sum_{i=1}^{n} \frac{1}{C_{vi}(2,2)}\left(c_y + \frac{1}{c_z}p_y u_{yi} - v_{yi}\right)^2 \tag{7}$$

n is the number of corner features
u_{xi} and u_{yi} are the model coordinates of the i^{th} corner feature when the object is viewed from a nominal range.
v_{xi} and v_{yi} are computed coordinates for the i^{th} corner.
For the purpose of illustration, it is assumed that the C_{vi} are diagonal.
p_x and p_y account for camera calibration and the nominal range of object.

The first summation corresponds to horizontal positioning of the object and the second summation corresponds to vertical positioning of the object. The range to the object c_z enters into both summations. For example, the term, $1/c_z * p_x$, can be thought of a scale factor that is a function of range. Thus in each summation, the center of the object and the inverse of range are determined such that the computed feature locations are brought into correspondence with the object model. The object model consists of feature locations given at a nominal range. The weighting by the inverse of variance in feature location will then give the best linear unbiased estimate (BLUE) [11] in the object's position.

To solve for the coordinates of the object position, (7) can be rewritten in matrix form:

$$J_l = (b_f - A_f x_f)^T C_f^{-1}(b_f - A_f x_f) \quad b_f = \begin{bmatrix} v_{x1} \\ v_{y1} \\ . \\ . \\ . \\ v_{xn} \\ v_{yn} \end{bmatrix} \quad A_f = \begin{bmatrix} 1 & 0 & p_x u_{x1} \\ 0 & 1 & p_y u_{y1} \\ . & . & . \\ . & . & . \\ . & . & . \\ 1 & 0 & p_x u_{xn} \\ 0 & 1 & p_y u_{yn} \end{bmatrix} \quad C_f = \begin{bmatrix} C_{v1} & 0 & 0 & 0 & 0 \\ 0 & . & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & C_{vn} \end{bmatrix} \quad x_f = \begin{bmatrix} c_x \\ c_y \\ \frac{1}{c_z} \end{bmatrix} \tag{8}$$

The solution for x_f that minimizes J can be expressed by:

Figure 4. Target with bookshelf in background

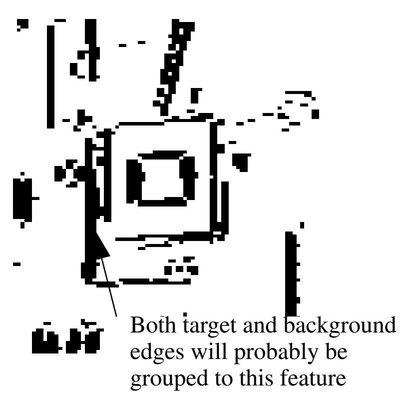

Figure 5. An edge image of target with bookshelf in background

$$\hat{x}_f = Lb_f \qquad L = (A_f^T C_f^{-1} A_f)^{-1} A_f^T C_f^{-1} \tag{9}$$

2.9. Confidence in object location

The reliability of the computed target location will vary from image to image. Reliability will depend on how well the object's features are discriminated.

Since object location is a linear function of feature location, covariance in object location could be propagated as in (6) by:

$$C_l = LC_f L^T \tag{10}$$

However, C_f is derived from an empirical estimate of the variance based on the spread of edge points around each feature. It is not the true statistical variance in each measured feature location. Moreover, it is based strictly on image data in the local neighborhood of the feature. It does not account for how much each feature contributes to the overall least squares error in the object location (i.e., b_f is completely absent from the formulation).

A better expression for the variance in feature location can be determined by accounting for the global consistency of all computed feature locations. The global consistency of the features can be measured by the normalized least squares error in the target's location:

$$e = \frac{(C_f^{-1/2})(b - A\hat{x})}{trace(C^{-1/2})} \tag{11}$$

This is a measure of the overall accuracy in fitting the target model to the conglomerate of feature locations. Each component of e, e_j, measures the consistency of the j^{th} feature measurement to the overall object location as computed by $\hat{x}$. Using this information, the covariance in the target's location can be approximated by:

$$C_l = LC_{fe}L^T \qquad C_{fe} = \begin{bmatrix} e_1^2 & 0 & \dots & 0 \\ 0 & e_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & e_n^2 \end{bmatrix} \tag{12}$$

2.10. Update of motion model and prediction

Using the computed target location, parametric models for the target motion are updated. Target motion is represented by separate 2nd order models along each coordinate direction. For example, motion along the x coordinate is represented by:

$$c_x[t] = a_{cx0} + a_{cx1}t + a_{cx2}t^2 \tag{13}$$

t = time

After each measurement of target location, the computed x coordinate, c_x, is used to update the parameters of the model, a_{cx0}, a_{cx1}, a_{cx2}. a_{cx0} represents position. a_{cx1} represents velocity. a_{cx2} represents 0.5 * acceleration. These parameters are updated by fitting the model to the time history of c_x. A weighted least squares fit with exponential decay is used. Each observation is weighted by the inverse of its covariance. In addition, the exponential decay gives increasingly less weight to older data. More specifically, the parameters (a_{cx0}, a_{cx1}, a_{cx2}) are determined by minimizing:

$$\sum_{n=0}^{t} \frac{1}{C_l(1,1)} (c_x[n] - (a_{cx0} + a_{cx1}n + a_{cx2}n^2))^2 \lambda^{t-n} \tag{14}$$

λ is the exponential decay factor: $0 < \lambda < 1$

The procedure for updating the other two models corresponding to c_y and c_z is identical.

The computed model parameters are then used to predict the location of the target in the next image and at the time of the next servo cycle (in Figure 1 this is represented by the arrows coming out of (6a) and (6b) respectively). For instance, the predicted x coordinate in the next image is:

$$\bar{c}_x[t+1] = a_{cx0} + a_{cx1}(t+1) + a_{cx2}(t+1)^2 \tag{15}$$

Using the prediction of object location, it is a straightforward computation to determine where the features of the object will fall on the image plane. These predicted feature locations are then used by the data association algorithm (processing stage (2) in Figure 1).

To solve for the model parameters in (14), the square root information filter algorithm is used. This is a recursive method; that is, each time there is a new data point, the solution is computed using the previous solution and the covariance in the previous solution. For a complete description of how the square root information filter algorithm is implemented to solve weighted least squares with exponential decay see [8].

In the frequency domain, the relationship between predicted and measured values, $\bar{c}_x$ and c_x respectively, (prediction of 1 cycle and $w_x(n) = 1$ for all n) is given by [12]:

$$\frac{\bar{c}_x(z)}{c_x(z)} = \frac{(-1+\lambda)(3 - 3z^{-1} - 3\lambda z^{-1} + z^{-2} + \lambda z^{-2} + \lambda^2 z^{-2})}{(-1+\lambda z^{-1})^3} \tag{16}$$

The frequency domain characteristics of this filter for the case of one sample period of prediction are illustrated in Figure 6, Figure 7, and Figure 8. The most salient characteristics of these plots are magnitude overshoot at high frequencies and frequency-dependent prediction time.

It can also be shown that this transfer function (16) is equivalent to an α–β–γ filter [12]:

$$H(z) = \frac{\left(\alpha+\beta+\frac{\gamma}{4}\right) + \left(-2\alpha-\beta+\frac{\gamma}{4}\right)z^{-1} + \alpha z^{-2}}{1 + \left(\alpha+\beta+\frac{\gamma}{4}-3\right)z^{-1} + \left(-2\alpha-\beta+\frac{\gamma}{4}+3\right)z^{-2} + (\alpha-1)z^{-3}} \tag{17}$$

Equivalency can be achieved with:

$$\alpha = 1-\lambda^3 \qquad \beta = 1.5 - 1.5\lambda - 1.5\lambda^2 + 1.5\lambda^3 \qquad \gamma = 2 - 6\lambda + 6\lambda^2 - 2\lambda^3 \tag{18}$$

For a more complete discussion of the frequency domain characteristics of α–β–γ filters refer to [14].

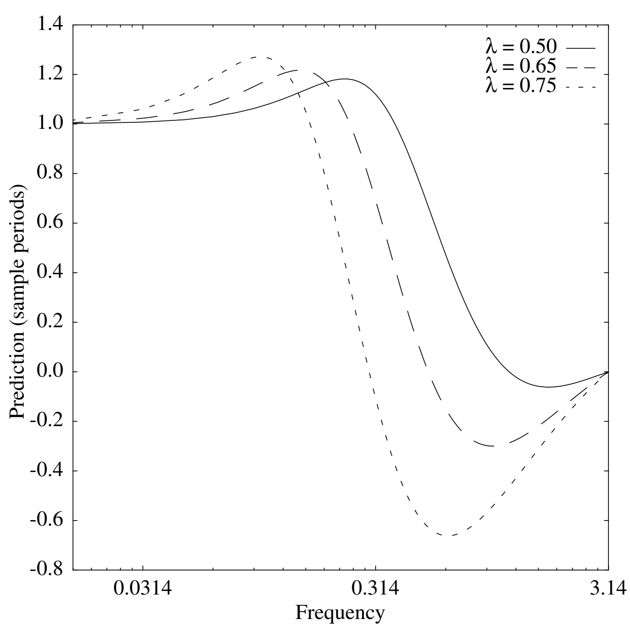

Figure 6. Prediction response of LS with exponential decay

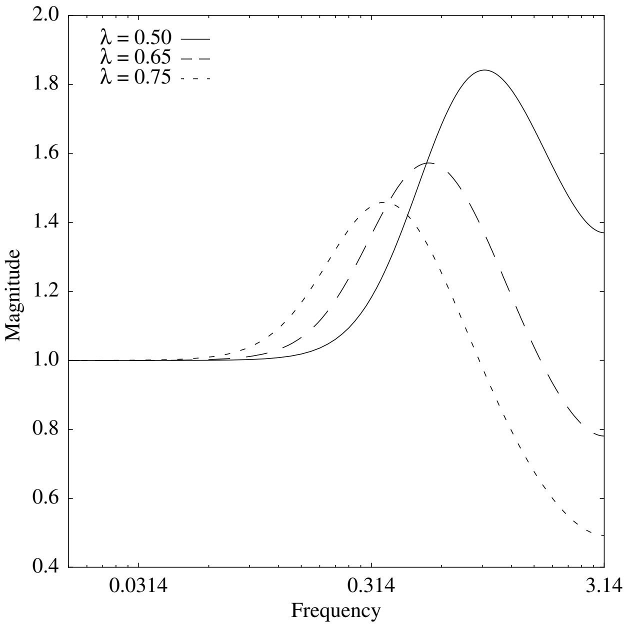

Figure 7. Magnitude response of LS with exponential decay

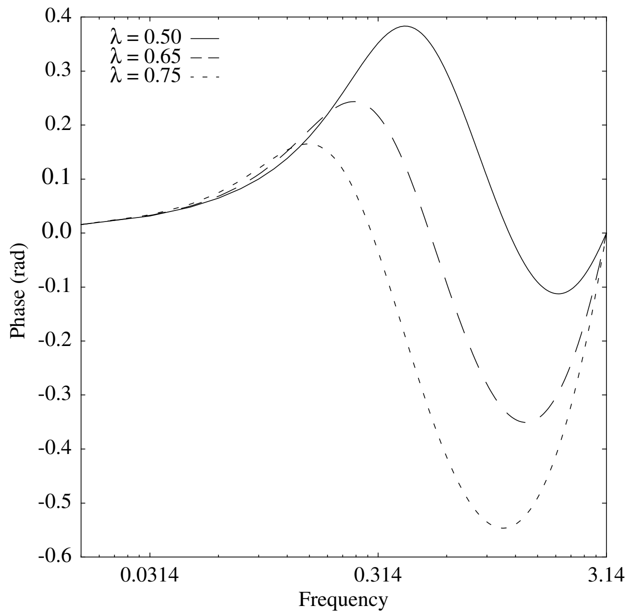

Figure 8. Phase response of LS with exponential decay

2.11. Servo control

Once prediction has been applied to the estimated target position to help compensate for image processing latency, this predicted position may be used as the command input to a servo-controlled system to enable the target to be actively followed. As discussed in the following sections, the predicted target position has been used as the input to a vehicle steering system to allow automated following of a lead vehicle, and as the input to an active vision system to allow a hand-held target to be tracked as it is moved about the room.

3.0. Experimental results

The computing system used for implementing the processes shown in Figure 1 consists of a PIPE[1] image processor for edge detection, a VME system with 4 68030 microprocessor boards for the remainder of the visual processing, and a second VME system for servo control of the active vision system. Program development and cross-compiliation were done on a Sun Sparc II. All code was programmed in the Ada programming language except for the programs controlling PIPE which were programmed in ASPIPE [15]. All code was written in accordance with the NASREM/RCS architecture [16].The system has an update rate of 30 hertz. The worst case delay between image acquisition and control actuation is approximately 100 ms. This target tracking approach was tested for two situations: vehicle following and tracking a hand-held target using an active vision system.

1. Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

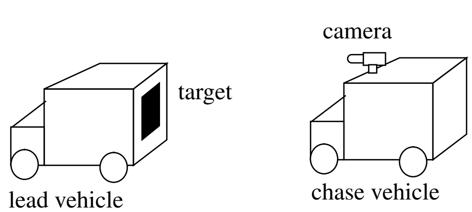

Figure 9. Vehicle following

3.1. Vehicle following

In vehicle following, or convoy driving, the goal is to follow the path of the vehicle driving in front your vehicle. In order to do this, the lead vehicle must be visually tracked from the vantage of a pursuing chase vehicle as illustrated in Figure 9. A photograph of the actual lead vehicle used in experiments is shown in Figure 2. This algorithm has been used to autonomously steer the NIST HMMWV [17][18] while pursuing this truck. Testing was performed on roads on the grounds of NIST. Successful vehicle following was achieved while traversing curves, turns at several intersections, and moderately shadowed roads. Top speeds of 35 km/h were achieved. Following distances varied from 5 to 15 meters.

3.2. Target tracking using TRICLOPS

The algorithm has also been extensively tested in the laboratory using TRICLOPS [13], a high-performance camera pointing system developed at NIST (Figure 10). An example image, Figure 5, was obtained from the right vergence camera of TRICLOPS during target tracking. Since this is a monocular tracking algorithm, only the right camera of TRICLOPS is used.

In these tests the system was able to track and servo on the target as it moved at velocities of up to 1 m/s at a distance of 0.85 m. The equivalent angular velocity for this tracking speed is about 1.2 rad/s (69 deg/s). The algorithm has demonstrated promising results in tracking the target in the presence of partial occlusion. The algorithm also was not confused by any of the background scenery it encountered. For earlier versions of the algorithm (using different confidence measures), tracking was unreliable when the target passed in front of a bookcase which created many horizontal and vertical edges. The algorithm is also fairly robust to deviations from the assumed fixed orientation of the target. The algorithm begins to fail when there are errors greater than 15 degrees in orientation. The algorithm will also fail if the target is moved with very sudden motions.

Further details regarding the performance specifications of TRICLOPS and the implementation of its motion control system, as well as additional discussion of predictive filter characteristics, may be found in [13] and [14].

4. Conclusion and future work

This paper has described a visual tracking algorithm. The primary features are maximal use of image information, confidence measures for feature aggregation and temporal aggregation, and prediction of target motion. This algorithm has demonstrated very promising results in tracking in the presence of partial occlusion and in the presence of potentially confusing backgrounds. Progress ic currently being made to extend this algorithm to tracking objects moving with a full 6 degrees of freedom of motion. Also, alternatives to edge detection may be considered in the future for feature detection.

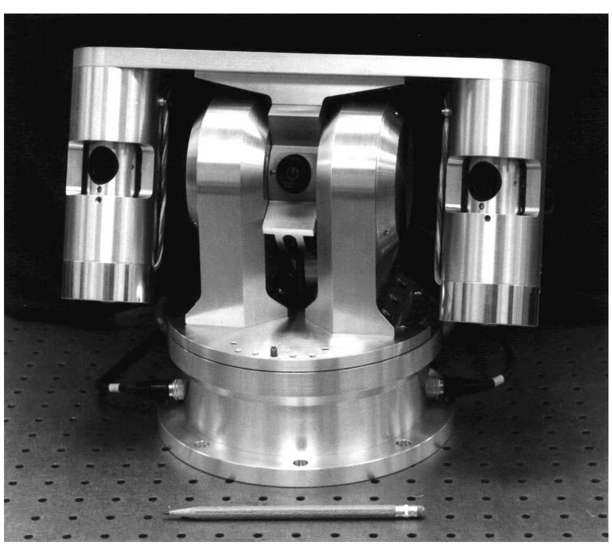

Figure 10. TRICLOPS active vision system

References

[1] A. Pentland, B. Horowitz, S. Sclaroff. “Non-Rigid Motion and Structure from Contour.” IEEE Workshop on Visual Motion. Oct. 7-9, 1991. Princeton, N. J.

[2] D. B. Gennery. “Visual Tracking of Known Three-Dimensional Objects.” International Journal of Computer Vision. 7:3, pp. 243 - 270. 1992.

[3] E.D Dickmann and V. Graffe. “Dynamic monocular machine vision.” Machine Vision Applications. 1: pp. 223-240. 1988.

[4] E.D Dickmann and V. Graffe. “Applications of dynamic monocular machine vision.” Machine Vision Applications. 1: pp 241-261. 1988.

[5] D. G. Lowe. “Robust Model-based Motion Tracking Through the Integration of Search and Estimation.” International Journal of Computer Vision. 8:2. pp. 113-122. 1993.

[6] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. “Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision.” IEEE Transactions on Robotics and Automation. 9:1. pp. 14-35. 1993.

[7] G. C. Goodwin, Kwai Sang Sin. *Adaptive Filtering, Prediction, and Control*. Prentice-Hall, Inc. 1984.

[8] H. Schneiderman, M. Nashman, R. Lumia. “Model-based vision for car following.” SPIE’s International Symposium on Optical Tools for Manufacturing and Advanced Automation. SPIE vol. 2059. 1993.

[9] Helstrom, C. W. *Probability and Stochastic Processes for Engineers*. Macmillan Publishing Company. 1984,

[10] Papoulis, A. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill. 1984.

[11] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, Inc. Englewood Cliffs, New Jersey. 1993.

[12] H. Schneiderman, A. J. Wavering. “Equivalency of LMSF and a-b-g filters.” In publication.

[13] A. J. Wavering, J. C. Fiala, K. J. Roberts, R. Lumia, “TRICLOPS: A High-performance Trinocular Active Vision System.” In Proc. IEEE Conference on Robotics and Automation, Vol. 3, pp. 310-317, Atlanta, GA, 1993.

[14] A. J. Wavering, R. Lumia. “Predictive visual tracking.” In Proc. Intelligent Robots and Computer Vision XII: Active Vision and 3D Methods. SPIE vol. 2056. pp. 86-97. Boston, MA. September 8-9, 1993.

[15] Aspex Inc. “PIPE--An Introduction to the PIPE System.” New York, 1987.

[16] J. Albus, H. G. McCain, R. Lumia. “NASA/NBS Standard Reference Model for Telerobotic Control System Architecture (NASREM).” NIST Technical Note 1235. Gaithersburg, MD, July, 1987.

[17] S. Szabo, H. Scott, K. Murphy, S. Legowik, R. Bostelman. “High-Level Mobility Controller for a Remotely Operated Land Vehicle.” Journal of Intelligent and Robotic Systems. Vol. 5, pp 63-77. 1992.

[18] K. Murphy. “Navigation and Retro-Traverse on a Remotely Operated Vehicle.” Proceedings of the IEEE Conference on Intelligent Control and Instrumentation. Singapore, February, 1992.