Vision-Based Robotic Convoy Driving

H. Schneiderman[1,3], M. Nashman[2], A. J. Wavering[1], R. Lumia[1]

1. Building 220, Room B127; Intelligent Systems Division; National Institute of Standards and Technology; Gaithersburg, MD 20899
2. Building 220, Room B123; Intelligent Systems Division; National Institute of Standards and Technology; Gaithersburg, MD 20899
3. Current address: Robotics Institute; Carnegie Mellon University; Pittsburgh, PA 15213-3891.
Correspondence to: H. Schneiderman; Robotics Institute; Carnegie Mellon University; Pittsburgh, PA 15213-3891.

Abstract. This article describes a method for vision-based autonomous convoy driving. In autonomous convoy driving, a robotic vehicle autonomously pursues another vehicle. In our method, pursuit is achieved by visual guidance, by visually tracking a target mounted on the back of the pursued vehicle. Visual tracking must be robust since a failure will lead to catastrophic results. To make our system as reliable as possible, several techniques are used. In particular, the quality of the extracted features are taken into account and uncertainty measures are entered into all computations including a best linear unbiased estimate (BLUE) of the target position in each separate image and a polynomial least mean square fit (LMSF) to the time history of the target position to estimate the target's motion parameters. In actual experiments, robust autonomous convoy driving has been demonstrated in the presence of various lighting conditions including shadowing, other traffic, turns at intersections, curves and hills. A continuous autonomous convoy drive of over 20 miles was successful, at speeds averaging between 50 and 75 km/h.

Key words: convoy driving, caravan driving, autonomous navigation, mobile robots, visual tracking.

1. Introduction

In convoy driving, a lead vehicle is visually tracked from the vantage of a pursuing chase vehicle as shown in Figure 1. It is extremely important that the visual tracking algorithm is robust, since

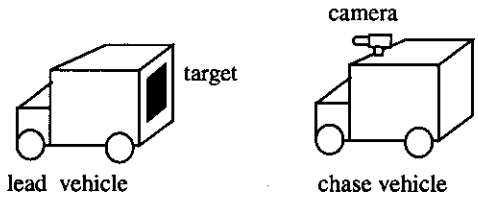

Figure 1. Autonomous convoy driving

the estimates of the lead vehicle position are used to autonomously steer the chase vehicle. In particular, the algorithm must maintain tracking in the presence of outdoor lighting conditions including various forms of shadowing on the target and roadway. As well, the algorithm must be able maintain tracking in the presence of a wide variety of background outdoor scenery including other vehicles, trees, buildings, etc.

To achieve reliability, our method is strongly motivated by statistical principles. It is recognized that low-level feature segmentation is not completely reliable. Individual features on the target may become confused with shadows and various entities in the background. These failures are typically localized in space and time. To reduce the influence of the local failures, our method strives to combine observations over space and time, with appropriate weighting to account for uncertainty, to obtain the best possible global estimate of the target's position and motion. The target's position in each image is computed using a best linear unbiased estimate (BLUE) from the individual feature locations. The motion parameters of the target are then estimated by computing a least mean polynomial fit (LMSF) to the time history of the target. Using this method, we have achieved robust visual tracking for the purposes of convoy driving including a continuous experimental run of over 20 miles.

Other methods that have been developed for autonomous convoy driving are somewhat limited in their performance. The system developed by Kehtarnavaz, et. al. [11] has been tested over short distances (on the order of a mile) and over a limited variety of roads and at maximum speeds of 20 mi/h. The system developed by Zielke, et. al [24][27] has been tested while the chase vehicle was driven manually. In one test, it was reported that during a 5 minute drive on the Autobahn, in a total of 1594 images, the pursued vehicle was correctly located 94.7% of the time. The system developed by Kories, et. al [13] has only been tested on videotaped image sequences. The method developed by Dickmanns, et. al. [6][7] has been tested in a limited fashion in which convoy driving was reported at speeds of approximately 5 m/s and over short distances and short durations of time (approximately 90 seconds).

In the broader context of visual target tracking, many methods have been developed and implemented [8][4][5][14][19][1][21][22]. With the exception of the ALVINN system for road following [21][22], none of these systems have reported sustained visual tracking in an outdoor environment comparable in duration to the method described in this paper.

2. Processing

An overview of the entire processing scheme is shown schematically in Figure 2. 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 - Extract 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 a model of the target

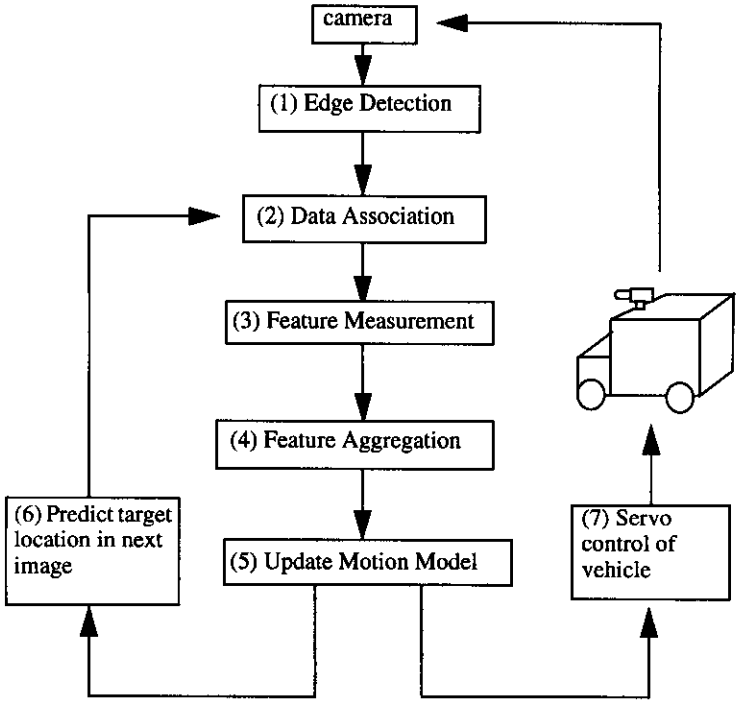

Figure 2. Processing Overview

to the conglomerate of computed feature locations using a BLUE estimate.

5. Update motion models - Update motion parameters (e.g. velocity, acceleration) of the object by computing a polynomial LMSF to the time history of the target.

6. 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. The predicted feature locations are then used for data association in the next image.

7. Servo control - Servo control of the steering on the autonomous vehicle such that it pursues the path of the lead vehicle. This is described in [25] [15] [17] .

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. The corners given by the intersection of the line segments are also maintained in the representation. One such model is shown in Figure 3 for the target shown in Figure 4.

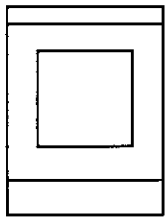

Figure 3. Model of target

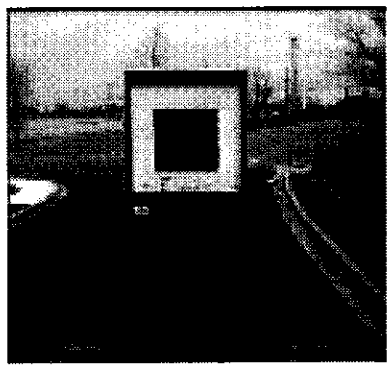

Figure 4. Target

2.2. Initialization of tracking

Registration between the model and the target is initially established by a teleoperator. Using a graphic representation of 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 directional 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. A list of the pixel locations of detected edges (i.e., those that satisfy the edge magnitude threshold) is compiled.

2.4. Data Association

Many of the detected edges 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.

The purpose of this processing is to determine which edge pixels best match each model line segment. In order for an edge pixel to be grouped to one of the line segments in the model, it must satisfy two criteria. The first criterion is the two-dimensional spatial proximity of the edge point to the predicted location of the line segment. (The predicted locations of the line segments are computed during the previous processing cycle from the prediction of the overall target location, (6) in Figure 2.) The second criterion is the similarity of direction of the edge with the predicted angular orientation of the line segment. An edge pixel is discarded if it does not satisfy both criteria for any line segment features.

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 against each line segment on a point by point basis for each thresholded edge point.

2.5. Feature Measurement: Line Features

After edge pixels have been grouped to each line segment feature, the overall location of the line feature is computed such that it gives the best fit to its group of associated edge pixels.

The location of each line segment feature is represented by two parameters *(m, b)*. One of two representations is chosen, depending on whether the feature is predicted to be oriented closer to the horizontal or vertical:

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

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

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

(x_i, y_i) are the coordinates of the i^{th} edge point.

n is the total number of edge points.

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

There are many methods that exist for minimizing the least squares residual in 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 line segment locations will be more accurate and reliable than others. The differ-

ences 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 excluded from the computation given by equation (3).

Uncertainty in the computed feature location can be approximated by an empirical measurement of variance:

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

A large variance corresponds to a scattered distribution of edge points about the computed feature location. Such a scattering of edge points usually indicates 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 or shadow edge) are grouped to a model feature.

2.7. Feature Measurement: Corner Features

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

$$w = F \quad \mathrm{v} \qquad w = \begin{bmatrix} -b_1 \\ -b_2 \end{bmatrix} \quad F = \begin{bmatrix} m_1 & -1 \\ m_2 & -1 \end{bmatrix} \quad v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{5}$$

Since corner position is a linear function of the line parameters, the covariance in v, C_v is then given by [9][20]:

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

2.8. Feature Aggregation

In this stage of processing, the overall target location is computed such that it gives the best fit to the conglomerate of computed corner feature locations. The target is 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 can be determined by finding (c_x, c_y, $1/c_z$) such that this weighted least squares residual, J_l, is minimized:

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

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 covariance matrices, C_{vi}, are diagonal. See eq. (8) for the general case.

p_x and p_y account for camera calibration and the nominal range of object used for u_{xi} and u_{yi}

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, $\frac{1}{c_z}p_x$, 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 corner locations given at a nominal range. The weighting by the inverse of variance in corner feature location will then give the best linear unbiased estimate (BLUE) [10] in the object's position.

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

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

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

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

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 and on the accuracy of the uncertainty measures (section 2.6).

Since object location is a linear function of feature location, covariance in object location, C_l, could be computed by propagating the covariance in feature location (similar to (6)) by:

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

However, C_f was derived from an empirical estimate of the variance and therefore is not necessarily an accurate measure of the true statistical variance in each measured feature location. Moreover, C_f 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, J_l, in the object location computed in (9) (i.e., b_f is completely absent from (10)).

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 least squares error in the target's location normalized by the trace of C_f:

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

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 empirical covariance in the target's location can be computed 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

For the purposes of tracking, motion along each coordinate direction can be considered a separate time sequence which can be filtered independently. One method of filtering a time sequence is to perform a least squares fit of the previous values of the sequence to polynomial function of time. Such a procedure is sometimes referred to as a polynomial least-mean-square-fit (LMSF) filter [18]. To fit a 2nd order polynomial to previous values of a time sequence, $v[t]$, a least squares residual, J, is minimized through choice of polynomial coefficients, a_0, a_1, a_2:

$$J = \sum_{n=0}^{t} w[n]\left(v[n] - \left(a_0 + a_1 n + \frac{a_2}{2}n^2\right)\right)^2 \lambda^{t-n} \tag{13}$$

$t \equiv$ *current time*

This residual incorporates two forms of weighting. Older data carries increasingly less influence, through use of an exponential decay profile, where older data is multiplied by larger powers of λ: $0 < \lambda \leq 1.0$. In actual experiments, we choose $\lambda = 0.77$ Each datum also carries an individual weight given by $w[n]$. For each coordinate direction, this is chosen to be the inverse of the computed covariance. For example, in filtering the x coordinate, $w = \frac{1}{c_l(1,1)}$.

The computed parameters a_0, a_1, a_2 represent the target position, velocity and acceleration, respectively, in one coordinate direction. These motion parameters are then used to predict the location of the target in the next image. For example, the predicted value of $v[t+1]$ is given by:

$$\bar{v}[t+1] = a_0 + a_1(t+1) + \frac{a_2}{2}(t+1)^2 \tag{14}$$

Given the predicted location of the object, the predicted locations of each individual feature are determined. These predicted feature locations are then used by the data association algorithm to process data from the next image as described in 2.4.

To solve for the model parameters in (13), a_0, a_1, a_2, the square root information filter algorithm is used [3]. This is a recursive method; that is, each time there is a new data point, the solution is recomputed using only this new datum explicitly. All past data are completely represented by the previous solution and the estimated covariance in the previous solution.

In the frequency domain, the relationship between predicted and measured values, $\bar{v}[t]$ and $v[t]$ respectively, (prediction of 1 cycle and $w_x(n) = 1$ for all n) is given by the following transfer function (see [23] for derivation):

$$\frac{\bar{v}(z)}{v(z)} = \frac{(-1+\lambda)(3-3z^{-1}-3\lambda z^{-1}+z^{-2}+\lambda z^{-2}+\lambda^2 z^{-2})}{(-1+\lambda z^{-1})^3} \tag{15}$$

The frequency domain characteristics of this transfer function for the case of one sample period of prediction are illustrated in Figure 5, Figure 6, and Figure 7.The most salient characteristics of

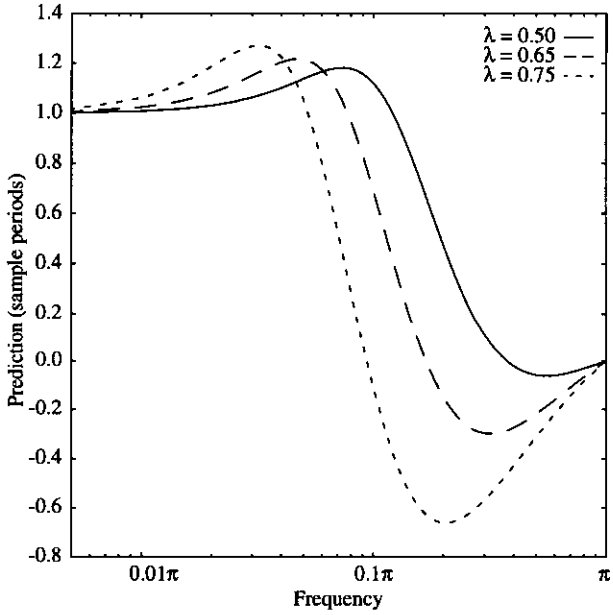

Figure 5. Prediction response of (15)

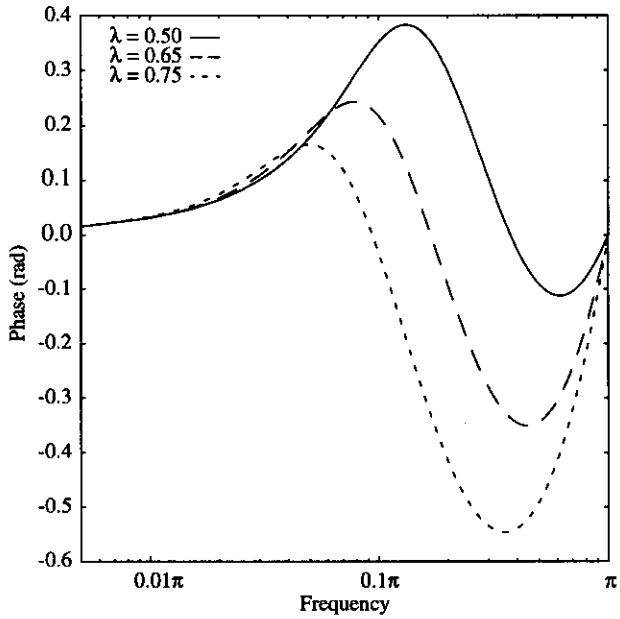

Figure 6. Phase response of (15)

these plots are magnitude overshoot at high frequencies and frequency-dependent prediction time.

It can also be shown that this transfer function (15) is equivalent to an α–β–γ filter [23]:

$$H(z) = \frac{\left(\alpha+\beta+\frac{\gamma}{4}\right)+\left(-2\alpha-\beta+\frac{\gamma}{4}\right)z^{-1}+\alpha z^{-2}}{1+\left(\alpha+\beta+\frac{\gamma}{4}-3\right)z^{-1}+\left(-2\alpha-\beta+\frac{\gamma}{4}+3\right)z^{-2}+(\alpha-1)z^{-3}} \tag{16}$$

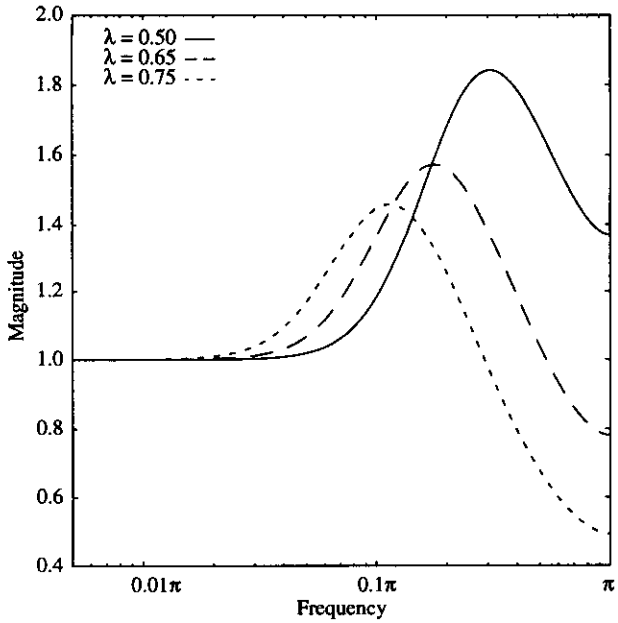

Figure 7. Magnitude response of (15).

Equivalency of (15) and (16) can be achieved [23] with:

$$\alpha = 1 - \lambda^3 \qquad \beta = 1.5 - 1.5\lambda - 1.5\lambda^2 + 1.5\lambda^3 \qquad \gamma = 2 - 6\lambda + 6\lambda^2 - 2\lambda^3 \qquad (17)$$

For a more complete discussion of the frequency domain characteristics of α–β–γ filters refer to [26].

3. Implementation and hardware

The complete autonomous vehicle system used for convoy driving consists of several major elements: a vehicle instrumented for computer control, a camera, and computer system as shown in Figure 8.

The vehicle used in these experiments was an Army High Mobility Multipurpose Wheeled Vehicle (HMMWV) customized for computer control [25] . This vehicle has computer controlled actuators to control steering, throttle, and brake. For the convoy driving experiments, only the steering was controlled autonomously, while brake and throttle were manually controlled. A standard CCD camera was mounted above the cab of the vehicle to provide video images to the computer system

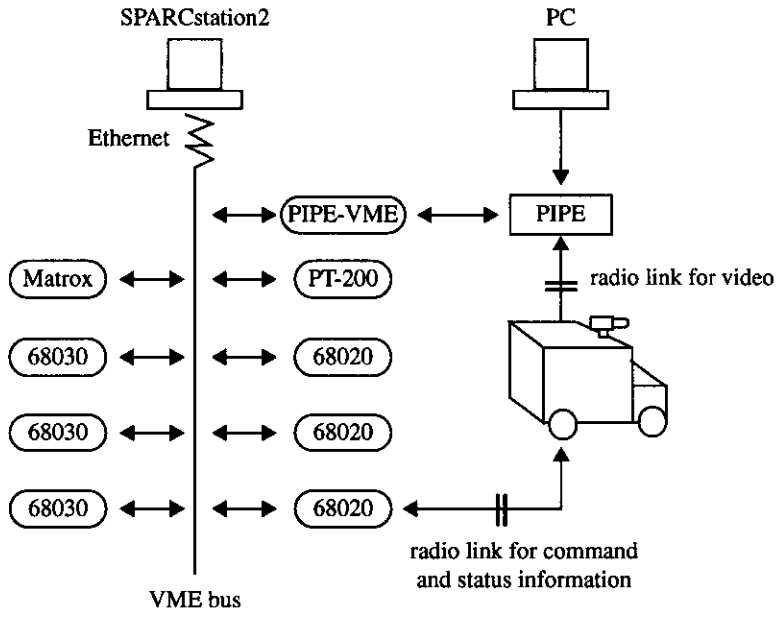

Figure 8. Hardware configuration

through a radio video link.

The computer system resides in a mobile computing and communications van [16] . The computer system consists of a Pipelined Image Processing Engine (PIPE), a VME multiprocessor system, and a Sun SPARCstation 2[1].

The incoming video images are supplied to PIPE. PIPE is a special purpose image processing engine that was originally developed at NIST [12]. The incoming video is digitized on PIPE to provide 242 x 256 images with 8 bit grey scale resolution. PIPE performs edge detection and produces a list of edge pixels for each image. This list is made available to the VME system via the PIPE-VME interface card. Program development for PIPE was done on a personal computer (PC) using the ASPIPE graphical programming language [2].

The remaining processing is divided among microprocessors connected on a VME bus. A 68020 processor is dedicated as a driver to the PIPE-VME interface card. This processor reads the edge list from the PIPE-VME interface card and copies it to globally accessible memory. Three 68030s in parallel preform the data association and feature measurement on different sets of features. The remainder of the vision processing is performed serially on a 68030 processor. A second 68020 performs computation for steering and communicates with the vehicle through a radio link. A third 68020 is used for the operator interface, data logging, and as a controller of a Matrox frame grabber. The Matrox frame grabber is used to create a graphic overlay showing the computed location of the target superimposed on to the live video image. Communication between microprocessors on the VME bus is achieved through semaphored common memory residing on a Performance Technologies' PT-200 2 Mbyte RAM board. Program development and cross-compiliation for the VME system is done on a Sun SPARCstation 2. All programming was done in the Ada language using the Verdix development and debugging system. Code is downloaded through ethernet and monitored through RS-232 connections from each microprocessor multiplexed to the SPARCsta-

1. Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify 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.

ence of a wide variety of background outdoor scenery including other vehicles, trees, buildings. Using the algorithm the vehicle was able to traverse a full variety of turns at various intersections, in which the target deviated as much as 20 degrees in orientation from its assumed fixed orientation. A continuous convoy drive over a 20 mile loop (approximately 33 km) was completed covering the NIST campus, and several roads in Gaithersburg and Germantown, Maryland, including Muddy Branch Road, Great Seneca Highway, Route 118, Key West Ave., and Darnestown Road. During this test, speed of the vehicle ranged from 30 to 45 miles/h (50 to 75 km/h) and following distances ranged from 5 to 15 meters. Other experiments showed that at following distances of greater than 20 meters, the target became too small in the image to be reliably tracked. A wide field of view lens was used to accommodate turns at intersections. This lens limited the resolution and thereby the ability to track targets at great distances. The algorithm also became susceptible to failure if the vehicle drove on very uneven, bumpy terrain causing large sudden perturbations of the camera or near dusk if the vehicle drove directly into the sun causing saturation of the image.

5. Summary and Conclusion

This article has described the vision processing for a system that has been successfully used for autonomous convoy driving. This algorithm was designed with the goal of achieving robust tracking. To achieve reliable performance, the method was motivated by statistical principles using a best linear unbiased estimate (BLUE) for target position and using a least mean polynomial fit (LMSF) to estimate the motion of the target. Using this method we have demonstrated autonomous convoy driving under a wide variety of testing conditions and demonstrated its reliability through a 20 miles continuous convoy drive over a variety of typical roads.

References

[1] Aloimonos, Y., Tsakiris, D. P. (1991). "On the Visual Mathematics of Tracking." *Image and Vision Computing*. (4)(9):235-251.

[2] .Aspex, Inc. (1987) "PIPE -- An Introduction to the PIPE System." New York.

[3] Bierman, G. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press. New York.

tion.

3.1. Timing

The sample rate for the system is 30 Hertz; that is, a new image is processed every 33.3 ms. The delay in prediction is one cycle; that is, the processing time from stages 2 through 6 in Figure 2 never exceeds 33.3 ms. The total time required for digitization and edge detection is 50.1 ms. The total elapsed time for computation between image acquisition and control actuation based on the image is approximately 84 ms in the worst case.

4. Experimental Results

This algorithm has been tested and used to autonomously steer the NIST HMMWV while pursuing a truck as shown in Figure 9. Testing has been performed on the grounds of NIST and nearby roads. Robust performance was demonstrated in the presence of various lighting conditions including shadowing on the target and on the roadway. The algorithm maintained tracking in the pres-

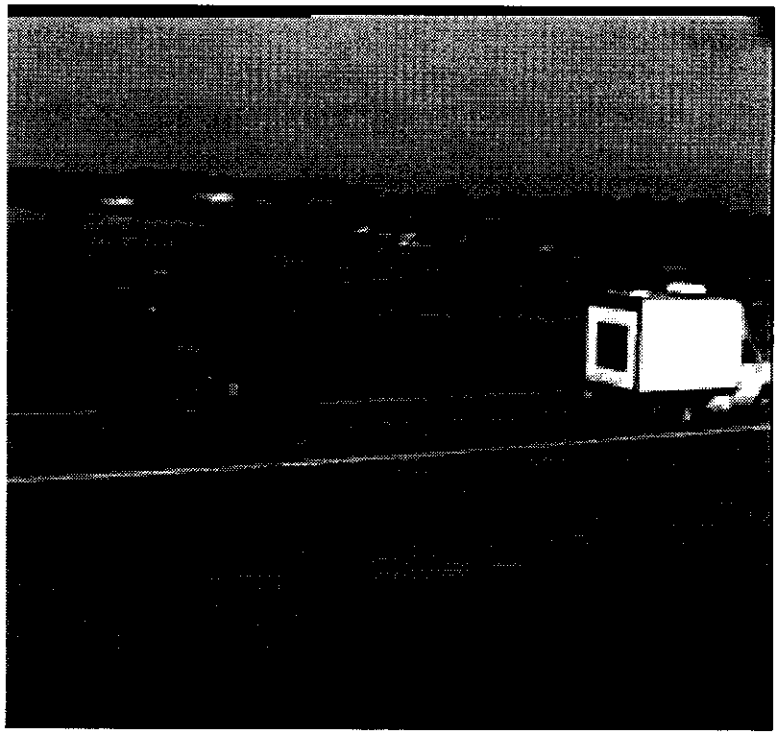

Figure 9. An autonomous convoy driving experiment

[4] Dickmanns, E. D. and Graffe, V. (1988a). "Dynamic monocular machine vision." *Machine Vision Applications*. 1:223-240.

[5] Dickmanns, E. D. and Graffe, V. (1988b). "Applications of dynamic monocular machine vision." *Machine Vision Applications*. 1:241-261.

[6] Dickmanns, E. D., Christians, T., Brudigam, C (1990). "Convoy Driving by Monocular Dynamic Vision." Pro-Art Workshop on Vision. Sophia-Antipolis.

[7] Dickmanns, E. D. (1991). "4-D Dynamic Vision for Intelligent Motion Control." *Engineering Applications of Artificial Intelligence*. 4(4):301-307.

[8] Gennery, D. B. (1992). "Visual Tracking of Known Three-Dimensional Objects." International Journal of Computer Vision. 7(3):243-270.

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

[10] Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, Inc. Englewood Cliffs, New Jersey.

[11] Kehtarnavaz, N., Griswold, N. C., Lee, J. S. (1991). "Visual Control of an Autonomous Vehicle (BART) - The Vehicle-Following Problem." *IEEE Transactions on Vehicular Technology*. 40(3):654-662.

[12] Kent, E. W., Shneier, M. O., Lumia, R. (1985). "PIPE." *Journal of Parallel and Distributed Computing*. 1985.

[13] Kories, R., Rehfeld, H., Zimmermann, G. (1988). "Toward Autonomous Convoy Driving: Recognizing the Starting Vehicle in Front." 9th International Conference on Pattern Recognition. 531-535. IEEE Computer Society Press.

[14] Lowe, D. G. (1993). "Robust Model-based Motion Tracking Through the Integration of Search and Estimation." *International Journal of Computer Vision*. 8(2):113-122.

[15] Murphy, K. N. (1992). "Navigation and Retro-Traverse on a Remotely Operated Vehicle." *Proceedings of the IEEE Conference on Intelligent Control and Instrumentation*. Singapore.

[16] Murphy, K. N., Juberts, M., Legowik, S. A., Nashman, M., Schneiderman, Scott, H. A., Szabo, S. (1993). "Ground Vehicle Control at NIST: from Teleoperation to Autonomous." SOAR '93. Houston, TX.

[17] Murphy, K. N. (1994). "Analysis of Robotic Vehicle Steering and Controller Delay." 5th International Symposium on Robotics and Manufacturing. Maui, Hawaii. August 15-17, 1994.

[18] Ng, L. C., LaTourette, R. A. (1983). "Equivalent bandwidth of a general class of polynomial smoothers." *Journal of Acoustic Society of America*. 74(3).

[19] Papanikolopoulos N. P., Khosla, P. K., Kanade, T. (1993). "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):14-35.

[20] Papoulis, A. (1984). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill. New York.

[21] Pomerleau, D. (1990). "Neural Network Based Autonomous Navigation." In *Vision and Nav-*

igation: The Carnegie Mellon Navlab. Kluwer. Norwell, MA.

[22] Pomerleau, D. (1992). "Progress in Neural Network-based Vision for Autonomous Robot Driving." *Proceedings of Intelligent Vehicles '92 Symposium*. 391-396.

[23] Schneiderman, H., Wavering, A. J. (1994). "Equivalency of LMSF and α–β–γ filters." In publication.

[24] Schwarzinger, M., Zielke, T., Noll, D., Brauckmann, M., von Seelen, W. (1992). "Vision-Based Car-Following: Detection, Tracking, and Identification." *Proceedings of the Intelligent Vehicles '92 Symposium*. pp 24 - 29. Detroit, Michigan. July, 1992.

[25] Szabo, S., Scott, H., Murphy, K. N., Legowik, S., Bostelman, R. (1992). "High-Level Mobility Controller for a Remotely Operated Land Vehicle." *Journal of Intelligent and Robotic Systems*. 5:63-77.

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

[27] Zielke, T., Brauckmann, M., von Seelen, W. (1992). "CARTRACK: Computer Vision-Based Car-Following." IEEE Workshop on Applications of Computer Vision. pp. 156 - 163. Palm Springs, California.