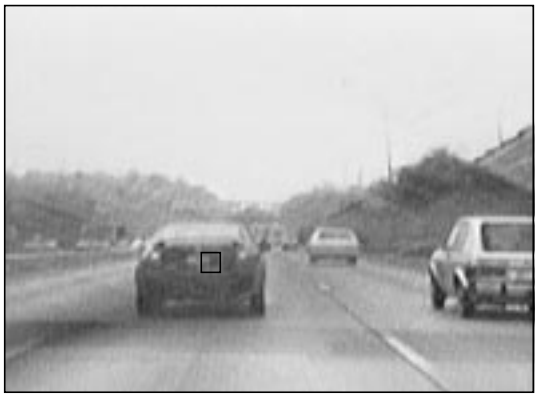
(b)

(d)



(c)

(e)

Figure 12 Vehicle tracking

(a)

(d)

(b)

(e)

Figure 11 Pedestrian tracking

(c)

(a)

The application of these techniques to two selected IVHS problems has demonstrated the feasibility of using a vision sensor to derive salient information from real imagery. The information may then be used as an input into intelligent control or advisory systems. In such a system, vision would constitute only one of the sensing modalities available to the system. Other sensors such as radar, laser and ultrasonic range-finders, infra-red obstacle detectors, GPS, Forward Looking Infra-Red (FLIR), microwave, etc. provide other modalities with particular strengths and weaknesses.

Issues for future research include the automatic selection of the feature window size (an issue discussed by Okutomi and Kanade[9]) in order to select a window that has some texture variations, simple model based techniques for distinguishing people or vehicles from background clutter, and the incorporation of the visual sensing data into larger, multisensor application systems for various transportation applications.

9. ACKNOWLEDGMENT

This work has been supported by the Department of Energy (Sandia National Laboratories) through Contract #AC-3752D, the National Science Foundation through Contract #IRI-9410003, the Center for Transportation Studies through Contract #USDOT/DTRS 93-G-0017-01, the 3M Corporation, the Graduate School of the University of Minnesota, and the Department of Computer Science of the University of Minnesota.

10. REFERENCES

1 P. Anandan, “A computational framework and an algorithm for the measurement of visual motion,” *International Journal of Computer Vision*, 2(3):283-310, 1988.

2 J.J. Craig, *Introduction to robotics: mechanics and control*, Addison-Wesley, Reading, MA, 1985.

3 E.D. Dickmanns, B. Mysliwetz, and T. Christians, “An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles”, *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1273-1284, November, 1990.

4 A. Houghton, G.S. Hobson, L. Seed, and R.C. Tozer, “Automatic monitoring of vehicles at road junctions,” *Traffic Engineering Control*, 28(10):541-453, October, 1987.

5 R.M. Inigo, “Application of machine vision to traffic monitoring and control,” *IEEE Transactions on Vehicular Technology*, 38(3):112-122, August, 1989.

6 N. Kehtarnavaz, N.C. Griswold, and J.S. Lee, “Visual control of an autonomous vehicle (BART) — the vehicle-following problem,” *IEEE Transactions on Vehicular Technology*, 40(3):654-662, August, 1991.

7 M. Kilger, “A shadow handler in a video-based real-time traffic monitoring system,” *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 11-18, 1992.

8 P.G. Michalopoulos, “Vehicle detection video through image processing: the autoscope system,” *IEEE Transactions on Vehicular Technology*, 40(1):21-29, February, 1991.

9 M. Okutomi and T. Kanade, “A locally adaptive window for signal matching,” *International Journal of Computer Vision*, 7(2):143-162, 1992.

10 N. 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):14-35, 1993.

11 N. Papanikolopoulos, “Controlled active vision,” Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992.

12 C. Pellerin, “Machine vision for smart highways,” *Sensor Revie*w, 12(1):26-27, 1992.

13 C. Smith, S. Brandt, and N. Papanikolopoulos, “Controlled active exploration of uncalibrated environments,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 792-795, 1994.

14 M. Takatoo, T. Kitamura, Y. Okuyama, Y. Kobayashi, K Kikuchi, H. Nakanishi, and T. Shibata, “Traffic flow measuring system using image processing,” *Proceedings of SPIE*, 1197:172-180, 1990.

15 C. Thorpe, M.H. Hebert, T. Kanade, and S.A. Shafer, “Vision and navigation for the Carnegie-Mellon NAVLAB,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*,10(3):362-373, 1988.

16 B. Ulmer, “VITA- an autonomous road vehicle (ARV) for collision avoidance in traffic,” *Proceedings of the Intelligent Vehicles ’92 Symposium*, 36-41, 1992.

17 T. Zielke, M. Brauckmann, and W. Von Seelen, “CARTRACK: computer vision-based car-following,” *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 156-163, 1992.

has been annotated to indicate the cause of the tracking problems, that in all cases were due to either momentary occlusion of the tracked feature by windshield wipers on the primary vehicle or extreme changes in the relative brightness and contrast of the tracked feature as a result of the vehicle passing under overpasses on the freeway. In all cases tracking was resumed immediately after the visual disturbance ended. In actual time this corresponded to less than a second of lost tracking per occurrence. Simple filtering techniques (e.g., Kalman filtering) would effectively remove virtually all perturbation of the tracking window resulting from such events.

Five selected frames of the test are presented in Figure 12(a) through Figure 12(e) at the end of this paper. The Figure 12(c) is taken just prior to the loss of tracking due to windshield wiper occlusion. The blur due to the wiper is just appearing in the lower left of the frame.

Figure 9 and Figure 10 show the results of tracking a different feature (the spoiler) on back of the same vehicle tracked in the previous example. Tracking performance in this example is considerably worse than in the previous example due to the poor SSD surface characteristics of the feature selected by the operator. The tracking exhibits problems similar to the previous example, but longer in duration and fails completely at about frame 2000. This failure corresponds to the algorithm finding a suitable feature match on the surface markings of the freeway.

Automatic feature selection techniques presented earlier in this paper would not have chosen this feature for tracking due to its poor SSD surface characteristics.

8. CONCLUSION

This paper presents robust techniques for visual sensing in uncalibrated environments for intelligent vehicle- highway systems applications. The techniques presented provide ways of recovering unknown environmental parameters using the Controlled Active Vision framework[11]. In particular, this paper presents novel techniques for the detection and visual tracking of vehicles and pedestrians.

For the problem of visual tracking, we propose a technique based upon earlier work in visual servoing[10,11] that achieves superior speed and accuracy through the introduction of several performance enhancing techniques. The method-specific optimizations also enhance overall system performance without affecting worst-case execution times. These optimizations apply to various region-based vision processing applications and, in our application, provide the speedup required to increase directly the effectiveness of the real-time vision system.

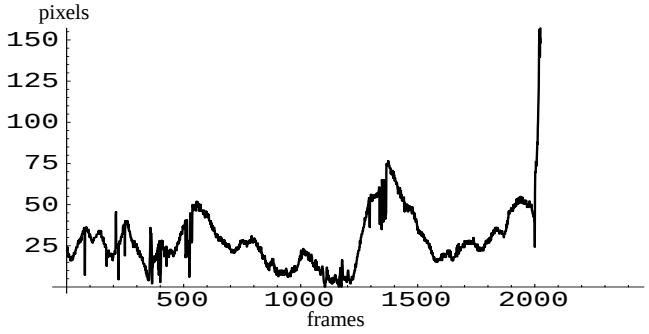


Figure 9 Vehicle tracking - reference point

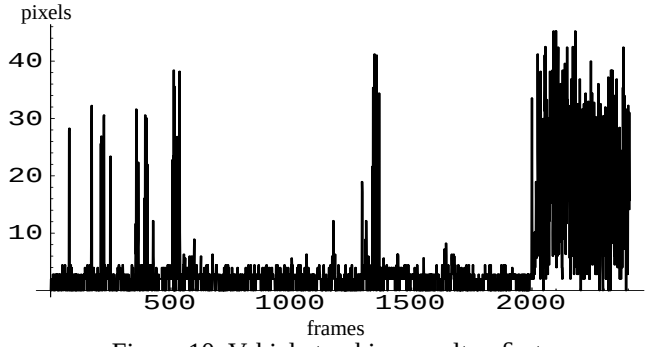


Figure 10 Vehicle tracking results - first

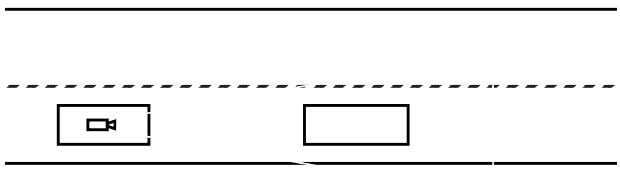


Figure 6 Experimental setup

The VPS includes a monitor upon which graphic representations of the tracking algorithm are displayed. While the system is tracking, the input data is displayed and a red box is drawn around the tracked location of the feature (see Figure 5). Five additional frames of the tracking appear at the end of this paper (see Figure 12(a) through Figure 12(e)). Initial experiments were done by watching the system track the features as they were displayed on the monitor. This allowed us to gain an understanding of the performance of the system under varying conditions and with a variety of features. In particular we discovered that the performance of the system is very sensitive to the robustness of the features selected.

After initial studies were completed, we proceeded to do a quantitative analysis of the vehicle tracking performance. While we were unable to compare the tracking results against ground-truth data (exact locations of objects/features manually calculated off-line), we were able to provide a reasonable determination of the system performance by assuming that the motion of the vehicle within the frame would be relatively smooth and within easily determinable velocity bounds. In other words, any sufficiently large, quick motions of the tracked location were likely to be the result of a loss of tracking rather than due to motion of the vehicle. This hypothesis was confirmed by viewing the tracking results and correlating the spikes in the plots of the tracked locations with the motion of the displayed tracking windows.

Figure 7 and Figure 8 show the results of one such experimental run where a single feature (the license plate) was tracked on the back of a single vehicle that was followed for approximately 2 minutes. Figure 7 shows the motion of the feature in the frame with respect to a reference point in the image that corresponds to an initial location of the feature. Figure 8 shows the first derivative of the plot in Figure 7. This plot clearly shows the relatively regular motion of the calculated feature position, corresponding to the relatively smooth motion of the tracked vehicle with respect to the vehicle carrying the sensor. The spikes in the plot correspond to the times where the tracking was lost and the tracking window rapidly moved off the feature. The plot

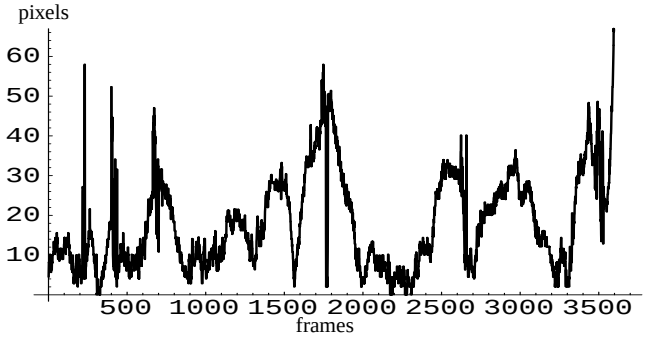


Figure 7 Vehicle tracking - reference point

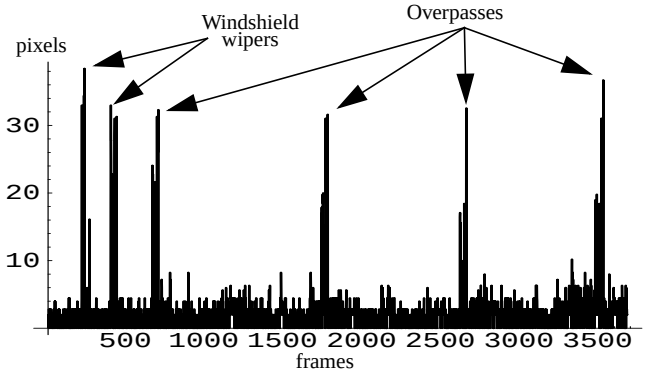


Figure 8 Vehicle tracking - first derivative

Figure 4 Pedestrian tracking

implemented. The method presented here may be used to track multiple features that in turn define the control points for an active deformable model (“snake”).

7.2. Vehicle tracking

A primary requirement of an IVHS system is that it must be able to detect and track potential obstacles. Of particular interest are other vehicles moving around and in front of the vehicle upon which the sensor is mounted (the primary vehicle). These vehicles constitute obstacles that must be avoided (collision avoidance) or a target that is to be followed (convoying).

Collision avoidance of moving vehicles (with a relatively constant velocity) detected near the primary vehicle can be effected with careful path planning. The basic goal is to maintain the desired speed and path (i.e., staying in the same lane of the road or highway) while avoiding other vehicles. Under normal circumstances this means accelerating and decelerating appropriately to avoid cars in front of and behind the primary vehicle. In extreme cases this means taking evasive action or warning the operator of a potential collision situation.

Another application area that involves the same basic problems is vehicle convoying. In this case, all path planning is done by the operator of the vehicle at the head of the convoy and all other vehicles must follow at a specified distance in a column behind the primary vehicle. For these reasons we chose to apply the Controlled Active Vision framework to the problem of tracking vehicles moving in roughly the same direction as the primary vehicle.

The experimental data was collected by placing a camcorder in the passenger seat of a car and driving behind other cars on the freeway (see Figure 6). This produced data that closely matched that which could be expected in a typical IVHS application. This data was later played back through a VCR and used as input to the VPS, which tracked the vehicles and produced a series of *(x,y)* locations specifying the detected locations of the features being tracked. Because the exact world coordinate locations of the features in each frame of the video are unknown, we were unable to provide a complete analysis of the tracking performance of the system. However, a simple assumption about the motion of the vehicle in the frame yields conclusive results that match the intuitive results gained from viewing the graphic display of the vehicle tracking algorithm.

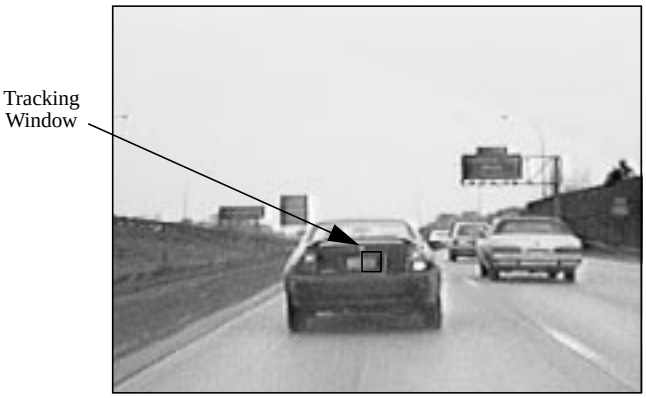


Figure 5 Vehicle tracking

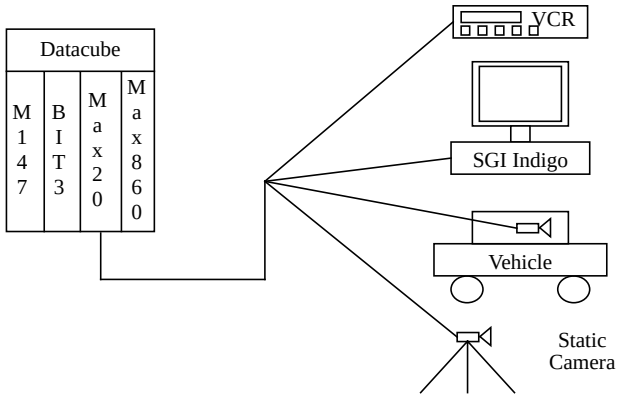


Figure 2 VPS system architecture

trol software. The video processing and calculations required to produce the desired control input are performed under a pipeline programming model using Datacube's Imageflow libraries.

7. EXPERIMENTAL RESULTS

7.1. Pedestrian tracking

Pedestrians constitute one class of trackable objects that are of extreme importance to many IVHS applications. Considering the potential for injury when pedestrians and vehicles interact harmfully, the ability to track pedestrians should be considered a mandatory element of any visual tracking system that is nominated as a potential sensor for IVHS applications. Pedestrian tracking was therefore selected as the first area to which our tracking paradigm was applied. In these experiments we consider the tracking of a pedestrian at an intersection by a static camera. Such tracking has direct application to intelligent signal control and to collaborative traffic/vehicle management systems.

The goal of the first experiments was to demonstrate that our methods could successfully track a pedestrian under normal environmental situations. The experiments consisted of a single pedestrian crossing a street at a controlled intersection (see Figure 3). The camera was mounted on the opposite side of the street in a position that was consistent with a mounting position on the utility pole supporting the intersection's crosswalk signals. Imagery was captured from a real intersection using a camcorder and was later input into the VPS using a video cassette recorder.

A single pedestrian crossed the street at the crosswalk, moving toward the camera (see Figure 4). Five more frames of the pedestrian tracking appear at the end of this paper (see Figure 11 (a) through Figure 11(e)). The system tracked a feature on the pedestrian (the contrast gradient at the pedestrian's waist). Target tracking was not lost during the crossing, in spite of the degraded contrast in the imagery due to an overcast sky.

The performance of this pedestrian tracking can be enhanced with the use of explicit representations of people as semi-rigid bodies that deform in known and predictable ways. Such tracking would allow the system to track pedestrians in spite of rapidly changing conditions (i.e., the pedestrian moving from shadow to bright light) and clutter in the image. Additionally, occlusion (due to a passing vehicle, other pedestrians, etc.) could be handled in a more robust manner than has been currently

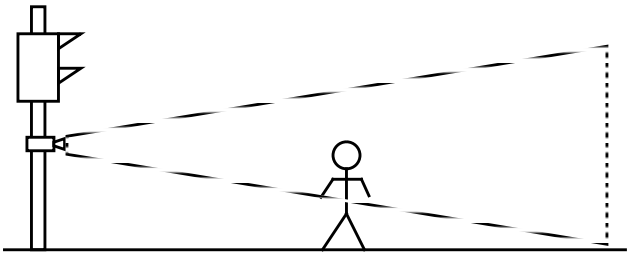


Figure 3 Experimental setup

Since the structure that implements the spiral search pattern contains no more overhead than the loop structures of the traditional search, worst-case performance is identical. In the general case, search time is approximately halved.

The third optimization arose from the observation that search times for feature points varied significantly, by as much as 100%, depending upon the shape/orientation of the feature. In determining the cause of the problem and exploring possible solutions, we realized that by applying the spiral image traversal pattern to the calculation of the SSD measure we could simultaneously fix the problem and achieve additional performance improvements. Spiraling the calculation of the SSD measure yields best-case performance that is independent from the orientation of the target by changing the order of the SSD calculations to no longer favor one portion of the image over another. In the traditional calculation pattern (a row-major traversal of the feature region) information in the upper portion of the region is used before that in the lower portion of the region, thus skewing the timing in favor of those images where the target or the foreground portion of the feature being tracked appears in the upper half of the region. Additional speed gains are achieved because the area of greatest change in the SSD measure calculations typically occurs near the center of the feature window, which generally coincides with an edge or a corner of the target, resulting in fewer calculations before the loop is terminated in non-matching cases. Speed gains from this optimization are approximately 40%.

When combined, these three optimizations (the loop short-circuiting, the spiral search pattern, and the spiral SSD calculation) interact cooperatively to find the minimum of the SSD surface as much as 17 times faster on average than the unmodified search.

Experimentally, the search times for the unmodified algorithm averaged 136 msec over 5000 frames under a variety of relative feature point motions. The modified algorithm with the search-loop short-circuiting alone averaged 60-72 msec search times over several thousand frames with arbitrary relative feature point motion. The combined short-circuit/spiral search algorithm produced search times that averaged 13 msec under similar tests and the combined short-circuit, dual-spiral algorithm produced search times that averaged 8 msec. Together these optimizations allow the vision system to track three to four features at RS-170 video rates (33 msec per frame) without video under-sampling.

5. FEATURE POINT SELECTION

In addition to the system latency and the effect of large displacements, an algorithm based upon the SSD technique may fail due to repeated patterns in the intensity function of the image or due to large areas of uniform intensity in the image. Both cases can provide multiple matches within a feature point's neighborhood, resulting in spurious displacement measures. In order to avoid this problem, our system automatically evaluates and selects feature points.

Feature points are selected using the SSD measure combined with an auto-correlation technique to produce an SSD surface corresponding to an auto-correlation in the neighborhood Ω[1,11]. Several possible confidence measures can be applied to the surface to measure the suitability of a potential feature point.

The selection of a confidence measure is critical since many such measures lack the robustness required by changes in illumination, intensity, etc. We utilize a two-dimensional displacement parabolic fit that attempts to fit parabola $e(\Delta x_r) = a\Delta x_r^2 + b\Delta x_r + c$ to a cross-section of the surface derived from the SSD measure[11]. The parabola is fit to the surface in several predefined directions. A feature point is selected if the minimum directional measure is sufficiently high, as measured by (6).

6. THE VISION PROCESSING SYSTEM

The Vision Processing System (VPS) used for these experiments is the image processing component of the Minnesota Robotic Visual Tracker (MRVT)[13] (see Figure 2). The VPS receives input from a video source such as a camera mounted in a vehicle, a static camera, or stored imagery played back through a Silicon Graphics Indigo or a video tape recorder. The output of the VPS may be displayed in a readable format or can be transferred to another system component and used as an input into a control subsystem. This flexibility offers a diversity of methods by which software can be developed and tested on our system. The main component of the VPS is a Datacube MaxTower system consisting of a Motorola MVME-147 single board computer running OS-9, a Datacube MaxVideo20 video processor, and a Datacube Max860 vector processor in a portable 7-slot VME chassis. The VPS performs the optical flow and calculates any desired control input. It can supply the data or the input via shared memory to an off-board processor via a Bit-3 bus extender for inclusion as an input into traffic or vehicle con-

$$e(\mathbf{p}(k-1), \Delta\boldsymbol{x}) = \sum_{m,\, n \in N} [I_{k-1}(x(k-1)+m, y(k-1)+n) - I_k(x(k-1)+m+u, y(k-1)+n+v)]^2 \quad (6)$$

where $u, v \in \Omega$, N is the neighborhood of $\mathbf{p}$, m and n are indices for pixels in N, and I_{k-1} and I_k are the intensity functions in images *(k–1)* and *(k)*.

The size of the window N must be carefully selected to ensure proper system performance. Too small an N fails to capture enough contrast while too large an N increases the associated computational overhead and enhances the background. In either case, an algorithm based upon the SSD technique may fail due to inaccurate displacements. An algorithm based upon the SSD technique may also fail due to too large a latency in the system or displacements resulting from motions of the object that are too large for the method to accurately capture. We introduce several optimization techniques in the following section to counter these concerns.

4.2. Search optimizations

The primary source of latency in a vision system that uses the SSD measure is the time needed to identify the minimizing $(u, v)^T$ in (6). To find the true minimum, the SSD measure must be calculated over each possible $(u, v)^T$. The time required to produce an SSD surface and to find the minimum can be greatly reduced by employing three schemes that, when combined, divide the search time significantly in the expected case.

The first optimization used is loop short-circuiting. During the search for the minimum on the SSD surface (the search for $(u, v)^T_{\min}$), the SSD measure must be calculated according to (6). This requires nested loops for the m and n indices. During the execution of these loops, the SSD measure is calculated as the running sum of the squared pixel value differences. If the current SSD minimum is checked against the running sum as a condition on these loops, the execution of the loops can be short-circuited as soon as the running sum exceeds the current minimum. This optimization has a worst-case performance equivalent to the original algorithm plus the time required for the additional condition tests. This worst case occurs when the SSD surface minimum lies at the last $(u, v)^T$ position searched. On average, this type of short-circuit realizes a decrease in execution time by a factor of two.

The second optimization is based upon the heuristic that the best place to begin the search for the minimum is at the point where the minimum was last found on the surface and to expand the search radially from this point. This heuristic works well when the disturbances being measured are relatively regular. In the case of tracking, this corresponds to targets that have locally smooth velocity, acceleration, and jerk curves. If a target's motion does not exhibit such relatively smooth curves, then the target itself is fundamentally untrackable due to the inherent latency in the video equipment and the vision processing system.

Under this heuristic, the search pattern in the (k) image is altered to begin at the point on the SSD surface where the minimum was located for the $(k-1)$ image. The search pattern then spirals out from this point, searching over the extent of u and v. This is in contrast with the typical indexed search pattern where the indices are increased in a row-major scan fashion. Figure 1 contrasts a traditional row-major scan and the proposed spiral scan where the center position corresponds to the position where the minimum was last found. This search strategy may also be combined with a predictive controller to begin the search for the SSD minimum at the position that the predictive aspect of the controller indicates is the possible location of the minimum.

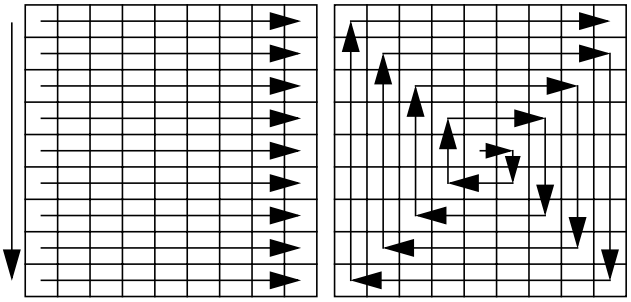


Figure 1 Traditional and spiral search patterns

positive" errors in which the figure image contains many pixels that don't necessarily belong to important objects. A smaller, more sensitive threshold can be used if the images are preprocessed with a low-pass filter. The filter spatially averages the pixels, making salt and pepper noise cause smaller difference values.The *figure images* are used for the automatic selection and tracking of features. In other words, instead of searching for features over the entire image, we automatically select features in areas where traffic objects of interest may exist.

4. VISUAL MEASUREMENTS

Our vehicle and pedestrian tracking applications use the same basic visual measurements that are based upon a simple camera model and a measure of optical flow in a temporal sequence of images. The visual measurements are combined with search-specific optimizations in order to enhance the visual processing from frame-to-frame and to optimize the performance of the system in our selected applications.

4.1. Camera model and optical flow

We assume a pinhole camera model with a world frame, R_W, centered on the optical axis. In addition, a focal length f is assumed. A point $\mathbf{P} = (X_W, Y_W, Z_W)^T$ in R_W, projects to a point $\mathbf{p}$ in the image plane with coordinates (x, y). We can define two scale factors s_x and s_y to account for camera sampling and pixel size, and include the center of the image coordinate system (c_x, c_y) given in image frame F_A[11]. This results in the following equations for the actual image coordinates (x_A, y_A):

$$x_A = \frac{fX_W}{s_x Z_W} + c_x = x + c_x \tag{1}$$

$$y_A = \frac{fY_W}{s_y Z_W} + c_y = y + c_y \tag{2}$$

Any displacement of the point $\mathbf{P}$ can be described by a rotation about an axis through the origin and a translation. If this rotation is small, then it can be described as three independent rotations about the three axes $\mathbf{X}_W$, $\mathbf{Y}_W$, and $\mathbf{Z}_W$[2]. We will assume that the camera moves in a static environment with a translational velocity (T_x, T_y, T_z) and a rotational velocity (R_x, R_y, R_z). The velocity of point $\mathbf{P}$ with respect to R_W can be expressed as:

$$\frac{d\mathbf{P}}{dt} = -\mathbf{T} - \mathbf{R} \times \mathbf{P}\,. \tag{3}$$

By taking the time derivatives and using (1), (2), and (3), we obtain:

$$u = \left[x\frac{T_z}{Z_W} - f\frac{T_x}{Z_W s_x}\right] + \left[xy\frac{s_y R_x}{f} - \left(\frac{f}{s_x} + x^2\frac{s_x}{f}\right)R_y + y\frac{s_y}{s_x}R_z\right] \tag{4}$$

$$v = \left[y\frac{T_z}{Z_W} - f\frac{T_y}{Z_W s_y}\right] + \left[\left(\frac{f}{s_y} + y^2\frac{s_y}{f}\right)R_x - xy\frac{s_x}{f}R_y - x\frac{s_x}{s_y}R_z\right]. \tag{5}$$

We use a matching-based technique known as the Sum-of-Squared Differences (SSD) optical flow[1]. For a point $\mathbf{p}(k-1) = (x(k-1), y(k-1))^T$ in the image $(k-1)$ where k denotes the kth image in a sequence of images, we want to find the point $\mathbf{p}(k) = (x(k-1)+u, y(k-1)+v)^T$. This point $\mathbf{p}(k)$ is the new position of the projection of the feature point $\mathbf{P}$ in image (k). We assume that the intensity values in the neighborhood N of $\mathbf{p}$ remain relatively constant over the sequence k. We also assume that for a given k, $\mathbf{p}(k)$ can be found in an area Ω about $\mathbf{p}(k-1)$ and that the velocities are normalized by time T to get the displacements. Thus, for the point $\mathbf{p}(k-1)$, the SSD algorithm selects the displacement $\Delta\mathbf{x} = (u, v)^T$ that minimizes the SSD measure

motion, target motion, or both. These measured displacements are then used as one of the inputs into an intelligent traffic advisory system.

Additionally, we propose a visual tracking system that does not rely upon accurate measures of environmental and target parameters. An adaptive filtering scheme is used to track feature points on the target in spite of the unconstrained motion of the target, possible occlusion of feature points, and changing target and environmental conditions. Relatively high-speed targets are tracked under varying conditions with only rough operating parameter estimates and no explicit target models. Adaptive filtering techniques are useful under a variety of situations, including the applications discussed in this paper: vehicle and pedestrian tracking.

First, we discuss some previous work and present our ideas on the detection of traffic objects. Then, we formulate the equations for visual measurements, including an enhanced SSD surface construction strategy and search optimizations. Next, we present a feature point selection scheme that automatically determines optimal features to be used for object tracking. Finally, we discuss results from feasibility experiments in both of the selected applications using the Controlled Active Vision framework.

2. PREVIOUS WORK

An important component of a real-time intelligent traffic advisory system is the acquisition, processing, and interpretation of the available sensory information regarding the traffic conditions. At the lowest level, the sensory information is used to derive discrete signals to drive the advisory system and at a higher level this information is used to study the patterns of traffic flow or to adjust the behavior of a global traffic advisory system. Information about traffic can be obtained through a variety of sensors such as loop detectors or vision sensors. Among them, the most commonly used is the loop detector. However, loop detectors provide local information and present significant errors in their measurements. Recently, many researchers have proposed computer vision techniques for traffic monitoring and vehicle control. Houghton *et al.*[4] have proposed a system for tracking vehicles at a road junction based on video images. Inigo[5] has presented a machine vision system for traffic monitoring and control. A system that counts vehicles based on video-images has been built by Pellerin[12]. Kilger [7] has done extensive work on shadow handling in a video-based real-time traffic monitoring system. Michalopoulos[8] has developed the Autoscope system for vision-based vehicle detection and traffic flow measurement. A system similar to the Autoscope traffic flow measuring system has been built by Takatoo *et al.*[14]. This system computes parameters such as vehicle average speed and spatial occupancy. A vision-based collision avoidance system has been proposed by Ulmer[16]. Zielke *et al.*[17] have developed the CARTRACK system that automatically selects the rear of vehicles in images and tracks them in real-time. In addition, similar car-following algorithms have been proposed by Kehtarnavaz *et al.*[6]. Finally, other groups[3,15] have developed vision-based autonomous vehicles.

3. DETECTION OF TRAFFIC OBJECTS

In order for an intelligent vision system to be able to robustly track traffic objects in unpredictable, real-world environments, it is required that the system has some means of detecting such objects automatically. In considering detection, it is helpful to view an image as a set of pixels that belong to one of two categories: *figure* or *ground*. Figure pixels are those which are believed to belong to a traffic object of interest, while ground pixels belong to the objects' environment. We consider detection to be the identification and the analysis of figure pixels in each image of the temporal sequence.

There is a wide variety of techniques that could be used for the identification of whether a pixel is part of the figure or ground. For example, we could possess a model of the average shape of automobiles and attempt to fit this model to locations within an image. However, identification schemes that are computationally intensive may not be able to complete detection in real-time. Using such schemes would cause the vision system to lack robustness. In searching for a fast means to estimate the figure/ground state of a pixel, we consider the heuristic that uninteresting objects (such as a sidewalk) tend to be displayed by pixels whose intensities are constant or very slowly changing over time, while objects of interest (such as a pedestrian) tend to be located where pixel intensities have recently changed. Thus, a comparison between images that occurred at different times may yield information about the existence of important objects.

The proposed scheme maintains a *ground image* that represents the past history of the environment. For each pixel in the current image, a comparison is made to the corresponding pixel in the ground image. If they differ by more than a threshold intensity amount, then the pixel is considered to be part of a binary *figure image*. If this threshold is too small, then portions of the object may blend into the background. If the threshold is too large, then slight changes in the environment will cause "false

Presented at SPIE's International Symposium on Photonics for Industrial Applications, Intelligent Vehicle Highway Systems, Oct 31 - Nov 4, Boston, MA, 1994.

Visual tracking strategies for intelligent vehicle-highway systems

Christopher E. Smith Nikolaos P. Papanikolopoulos Scott A. Brandt Charles Richards

University of Minnesota, Department of Computer Science
4-192 EE/CS Building
200 Union St. SE
Minneapolis, MN 55455

ABSTRACT

The complexity and congestion of current transportation systems often produce traffic situations that jeopardize the safety of the people involved. These situations vary from maintaining a safe distance behind a leading vehicle to safely allowing a pedestrian to cross a busy street. Environmental sensing plays a critical role in virtually all of these situations. Of the sensors available, vision sensors provide information that is richer and more complete than other sensors, making them a logical choice for a multisensor transportation system. In this paper we present robust techniques for intelligent vehicle-highway applications where computer vision plays a crucial role. In particular, we demonstrate that the Controlled Active Vision framework[11] can be utilized to provide a visual sensing modality to a traffic advisory system in order to increase the overall safety margin in a variety of common traffic situations. We have selected two application examples, vehicle tracking and pedestrian tracking, to demonstrate that the framework can provide precisely the type of information required to effectively manage the given situation.

Keywords: visual tracking, intelligent vehicle-highway systems, optical flow, pedestrian control, detection

1. INTRODUCTION

Transportation systems, especially those involving vehicular traffic, have been subjected to considerable increases in complexity and congestion during the past two decades. A direct result of these conditions has been a reduction in the overall safety of these systems. In response, the reduction of traffic accidents and the enhancement of an operator's abilities have become important topics in road safety. Improved safety can be achieved by assisting the human operator with a computer warning system and by providing enhanced sensory information about the environment. In addition, the systems that control the flow of traffic can likewise be enhanced by providing sensory information regarding the current conditions in the environment. Information may come from a variety of sensors such as vision, radar, and ultrasonic range-finders. The sensory information may then be used to detect vehicles, traffic signs, obstacles, and pedestrians with the objective of keeping a safe distance from static or moving obstacles and obeying traffic laws. Radar, Global Positioning System (GPS), and laser and ultrasonic range-finders have been proposed as efficient sensing devices. Vision devices (i.e., CCD cameras) have not been extensively used due to their high cost and noisy nature. However, the new generation of CCD cameras and computer vision hardware allows for efficient and inexpensive use of vision sensors as a component of a larger, multisensor system.

The primary advantage of vision sensors is their ability to provide diverse information on relatively large regions. Simple tracking techniques may be used with visual data taken from a vehicle to track several features of the obstacle ahead. This tracking allows us to detect obstacles (e.g., pedestrians, vehicles, etc.) and keep a safe distance from them. Optical flow techniques in conjunction with automatic selection of features allow for fast estimation of the obstacle-related parameters, resulting in robust obstacle detection and tracking with little operator intervention. In addition, surface features on the obstacles or knowledge of the approximate shape of the obstacles (i.e., the shape of the body of a pedestrian or automobile) may further improve the robustness of the tracking scheme. A single camera is proposed instead of a binocular system because one of our main objectives is to demonstrate that relatively unsophisticated and uncalibrated off-the-shelf hardware can be used to solve the problem. The ultimate goal of this research is to examine the feasibility of incorporating visual sensing into an automated system that provides information about pedestrians, traffic signs, and other vehicles.

One solution to these issues can be found under the Controlled Active Vision framework[11]. Instead of relying heavily on a priori information, this framework provides the flexibility necessary to operate under dynamic conditions where many environmental and target-related factors are unknown and possibly changing. The Controlled Active Vision framework utilizes the Sum-of-Squared Differences (SSD) optical flow measurement[1] as an input to a control loop. The SSD algorithm is used to measure the displacements of feature points in a sequence of images where the displacements may be induced by observer