Autonomous Robots 13, 87–104, 2002

A System for Semi-Autonomous Tractor Operations

ANTHONY STENTZ, CRISTIAN DIMA, CARL WELLINGTON, HERMAN HERMAN, DAVID STAGER
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract. Tractors are the workhorses of the modern farm. By automating these machines, we can increase the productivity, improve safety, and reduce costs for many agricultural operations. Many researchers have tested computer-controlled machines for farming, but few have investigated the larger issues such as how humans can supervise machines and work amongst them. In this paper, we present a system for tractor automation. A human programs a task by driving the relevant routes. The task is divided into subtasks and assigned to a fleet of tractors that drive portions of the routes. Each tractor uses on-board sensors to detect people, animals, and other vehicles in the path of the machine, stopping for such obstacles until it receives advice from a supervisor over a wireless link. A first version of the system was implemented on a single tractor. Several features of the system were validated, including accurate path tracking, the detection of obstacles based on both geometric and non-geometric properties, and self-monitoring to determine when human intervention is required. Additionally, the complete system was tested in a Florida orange grove, where it autonomously drove seven kilometers.

Keywords: agricultural robots, position estimation, path tracking, sensor fusion, obstacle detection

1. Introduction

Tractors are used for a variety of agricultural operations. When equipped with the proper implements, these mobile machines can till, plant, weed, fertilize, spray, haul, mow, and harvest. Such versatility makes tractors prime targets for automation. Automation promises to improve productivity by enabling the machines to drive at a higher average speed, improve safety by separating the human from the machine and minimizing the risk of accident, and reduce operational costs by minimizing the labor and maintenance needed for each machine.

Many researchers have investigated the automation of mobile equipment in agriculture. There are two basic approaches. In the first approach, the vehicle drives a route based on an absolute reference frame. The route is planned by calculating a geometric coverage pattern over the field or by manually teaching the machine. The route is driven by using absolute positioning sensors, such as a global positioning system (GPS), magnetic compass, or visual markers. The planned route is driven as programmed or taught, without modification. This approach is technically simpler, but it suffers from an inability to respond to unexpected changes in the field.

Using this approach, O'Conner et al. (1996) demonstrated an autonomous tractor using a carrier phase GPS with four antennae to provide both position and heading in the field. The system was capable of positional accuracy on the order of centimeters at straight-line speeds over 3 km/hour. Noguchi and Terao (1997) used a pair of static cameras in the field to track a visual marker on a tractor and triangulate its position. Additionally, the tractor was equipped with a geomagnetic direction sensor. The system was able to measure the tractor's position to an average error of 40 cm for fields up to 100 meters long. Erbach et al. (1991) used a pair of radio beacons to triangulate position in the field with errors of approximately 50 cm.

In the second approach, the vehicle drives a route based on a relative frame of reference. The route is planned by calculating a coverage pattern triggered by local reference cues, such as individual plants, a crop line, or the end of a row. The route is driven by using relative positioning sensors, such as a camera to detect crop rows or dead-reckoning sensors like odometry, accelerometers, and gyroscopes. The route is driven

as topologically planned, with exact position determined by the relative cues. This approach enables a machine to tailor its operation to individual plants as they change over time (e.g., applying more pesticides on larger trees), but it is technically more difficult.

Using this approach, Billingsley and Schoenfisch (1995) demonstrated an autonomous tractor driving in straight rows of cotton. The system used a camera with software for detecting the crop rows. It was able to drive on straight segments at 25 km/hour with a few centimeters of error. Gerrish et al. (1997) also used a camera to guide a tractor along straight rows. They reported results of 12 cm accuracy at speeds of 13 km/hour. Southall et al. (1999) used a camera to detect individual plants, exploiting the known planting geometry. These measurements were combined with odometry and inertial sensors. The system was used to guide an outdoor mobile robot for a short distance at a speed of about 2 km/hour.

More recently, researchers have investigated combining the two approaches. Ollis and Stentz (1996, 1997) and Pilarski et al. (1999) demonstrated an autonomous windrowing machine. The system used two modes of navigation: (1) a camera with crop line detection software; (2) a differential GPS integrated with odometry and a heading gyro. The windrower cut hundreds of acres of alfalfa fully autonomously at average speeds of 7 km/hour. The machine was able to cut swaths of crop and execute a spin turn at the end of the row. The system was steered by only one navigation mode at a time, with the other serving as a trigger or consistency check. Zhang et al. (1999) also used absolute and relative sensors, namely a camera, GPS, and heading gyro, to guide an autonomous tractor. The system used a rule-based method to process the sensor data and steer the machine. The system was able to drive the tractor at speeds of 13 km/hour with less than 15 cm of error.

Although full autonomy is the ultimate goal for robotics, it may be a long time coming. Fortunately, partial autonomy can add value to the machine long before full autonomy is achieved. For many tasks, the proverbial 80/20 rule applies. According to this rule, roughly 80% of a task is easy to perform and 20% is difficult. For the 80% that is easy, it may be possible for a computer to perform the task faster or more accurately (on average) than a human, because humans fatigue over the course of a work shift. The human can remain on the machine to handle the 20% of the task that is difficult. We call this type of semi-automation an *operator assist*. If the cost of the semi-automation is low enough and its performance high enough, it can be cost effective.

If the percentage of required human intervention is low enough, it may be possible for a single human to supervise many machines. We call this type of semi-automation *force multiplication*. Since the human cannot be resident on all machines at once, he/she must interact with the fleet over a wireless link.

The force multiplication scenario raises some challenging questions. How should operations be divided between human and machine? What is the best way for the human to interact with the machine? How can the machine be made productive and reliable for the tasks it is given? These are questions that address automation at a system level, that is, a system of people and computer-controlled machines working together. To date, the research literature has focussed primarily on achieving full autonomy, steadily improving the speed and reliability metrics, but without addressing the larger, system-level questions.

This paper explores these questions in the context of a semi-autonomous system for agricultural spraying operations in groves, orchards, and row crops. This application is well motivated, since spraying is performed frequently and is hazardous to both the vehicle operator and other workers in the field. We developed a computer-controlled John Deere 6410 tractor equipped with a GPS-based teach/playback system and camera-based obstacle detection. In April, 2000, we transported our tractor to an orange grove in Florida to collect data and test the autonomous navigation system. For one of the tests, the system was taught to drive a 7 kilometer path through the grove. It was then put in autonomous mode and drove the path at speeds ranging from 5 to 8 km/hour, spraying water enroute (see Fig. 1).

In this paper, we present an overview of the system and approach, and then we detail the user interface, navigation, and obstacle detection components. Finally, we summarize results and draw conclusions.

2. System Overview

2.1. Problem Characteristics and Tenets

In order to field semi-autonomous tractors in a force multiplication scenario, the machines must be productive, reliable, safe, and manageable by a human supervisor. In general, this is a very difficult problem to solve. Fortunately, for many tractor operations, we can

Figure 1. Semi-autonomous tractor spraying a grove.

take advantage of problem characteristics that simplify the task. From our analysis of the problem, we developed a system based on the following observations and tenets:

- *Take Advantage of Task Repetitiveness*: tasks like crop spraying, tillage, and planting entail driving up and down the same rows for each application. Given the repetition, the robotic tractors can be directly "taught" how to perform the task by observing a human operator. Furthermore, the tractor can learn and record the expected appearance of the terrain at each point along the route to detect obstacles or unexpected circumstances.
- *Err on the Side of False Positives*: when tracking a path or detecting obstacles, there are two types of mistakes that can be made: false positives and false negatives. For a false positive, the machine sees a problem where there are none. For a false negative, the machine fails to see a problem where there is one. Engineering the system to avoid one type of mistake increases the probability that the other type will occur. The ramifications of false negatives are more severe than false positives; therefore, we can guard against them by admitting more false positives, provided they are not so frequent that they severely disrupt the machine's operation.
- *Employ Humans as Remote Troubleshooters*: rather than dividing the tractor's tasks between human and computer, the computer is in control for all tasks with the human serving as a troubleshooter, namely classifying "positives" as true or false. Humans are quite good at quickly analyzing a potential problem and recommending a course of action. Short-term and infrequent human involvement is essential for implementing a successful force multiplication scheme.

2.2. *System Scenario*

For crop spraying, tractors apply the chemicals via a towed airblast or boom sprayer. The machines move slowly (e.g., 2 to 8 km/hour) up and down rows for each application. Periodically, the tractors drive to an irrigation ditch or rendezvous with a nurse tank to refill the chemical. The machines do not expect to encounter obstacles, but people and animals can move into harm's way, other vehicles can be left parked in the field, and gopher holes and other terrain hazards can suddenly appear.

For our complete solution, we envision a single human operator supervising a fleet of four or more tractors with sprayers. The tractors are equipped with positioning and obstacle avoidance sensors and communicate with the supervisor over a wireless radio link. Once per season (less often for tree applications), a human operator teaches the tractors the layout of the field by manually driving up and down the rows and along access routes. The operator presses buttons on a console to label route segments as "access route", "area to spray", etc. The tractor records this information along with positioning/speed data and scene appearance/range data for each route segment to build a complete "driving" map for the farm.

During execution, the system divides the task into portions for each tractor to execute. Each tractor drives a route segment by tracking positions and speeds stored in the map with its on-board positioning sensors. This process is assisted by a secondary navigation system

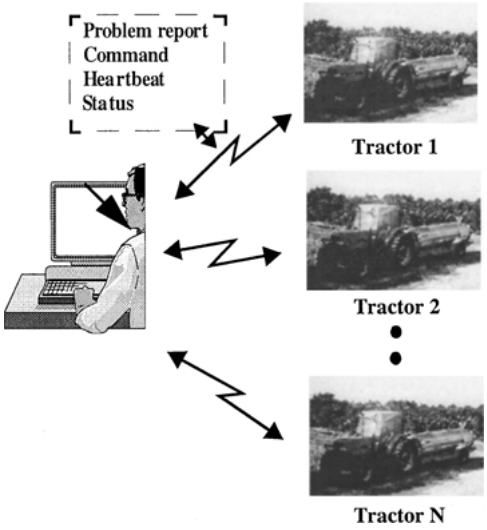

Figure 2. Human interface to fleet of tractors.

based on the tractor's visual sensors. While driving, each tractor sweeps the terrain in front with its sensors to check for obstacles. This process is guided by the appearance/range data stored in the route segment. When a tractor detects an obstacle in its path, it stops and transmits an image of the scene to the human operator (see Fig. 2). If the operator disagrees about the presence of an obstacle, or if the obstacle has moved out of the way, the operator signals the tractor to resume; otherwise, the operator travels to the stopped machine to rectify the problem. Periodically, the tractors deplete their chemical tanks and signal the operator. The operator meets the tractor at a rendezvous point, the tank is re-filled, and semi-autonomous operation is resumed.

We expect that this system will increase the productivity of each worker by at least a factor of four. Additionally, the tractors will be able to sense the trees and spray chemicals only where they are needed, thus reducing operational costs and minimizing negative environmental impact.

2.3. Experimental Test Bed

We have implemented many components of the full system described in the previous section. For our tests, we are using a computer-controlled, 90-horsepower Deere Model 6410 tractor (see Fig. 1). The tractor is equipped with a pair of stereo cameras for range and appearance data; a differential GPS unit, fiber optic heading gyro, doppler radar unit, and four-wheel odometry for positioning data; a pair of 350-Mz Pentium III processors running Linux for on-board processing, and a 1 Mbps wireless ethernet link for communication with the human operator. These sensors are described in more detail in the following sections.

Our approach is to develop the components necessary for a single tractor first, then replicate and integrate the components to produce a multi-vehicle system. Section 3 describes the operator console for training and supervising a semi-autonomous tractor. Section 4 describes the position-based navigation system for driving route segments. Section 5 describes the on-board obstacle detection and safeguarding. Section 6 describes conclusions.

3. Operator Console

In our scenario, the human operator interacts with the system during both the training phase and the operational phase. During training, the operator drives to a section of field and uses a teach interface to collect datapoints while manually driving a desired path. The interface shows the path as it is being recorded and displays the current status of the system. The system only allows paths to be recorded when it has a good position estimate. The operator also presses buttons during the teaching phase to label various parts of the path as "access route", "area to spray", etc. While teaching the path, the operator can pause, resume, and change parts of the path.

During full system operation, a single operator oversees a fleet of four tractors using a remote operator interface. This interface includes status information and live video from all four tractors, and it shows the tractors traveling down their paths on a common map. If any tractor has a problem, such as running out of chemical, encountering an obstacle, or experiencing a hardware failure, the interface displays a warning to the operator, shows pertinent information, and then he/she can take appropriate action to solve the problem. In the case of an obstacle, the system shows what part of the image was classified as an obstacle, and the operator decides whether it is safe for the tractor to proceed. To help determine if the path is clear, the tractor has a remotely-operated pan-tilt-zoom camera. If it is safe for the tractor to continue, the operator can simply click on a "resume" button and the tractor will continue its job.

We have implemented two interfaces that perform many of the capabilities described above. Figure 3 shows a simple teach interface that can record a path and display status information.

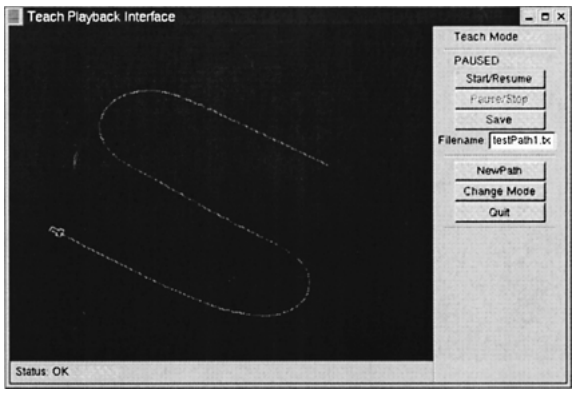

Figure 3. Teach interface. Operator interface showing the path that is being recorded and the status of the system while the system is being taught a path.

Figure 4 shows the remote interface. The goal of this interface is to allow both a simple global view of all the tractors during normal operation as well as a focused view that clearly shows all relevant information from a single tractor when there is a problem. Although only one tractor is currently operational, the interface is designed for four. In Fig. 4, the bottom left panel shows live video coming from the tractor's pan-tilt-zoom camera. The space to the right is reserved for video from the other tractors. This way, the operator will be able to see at a glance what all of the vehicles are doing. The upper half of the interface is switchable between focused views of each tractor and an overall map that shows all of the tractors together. Figure 4

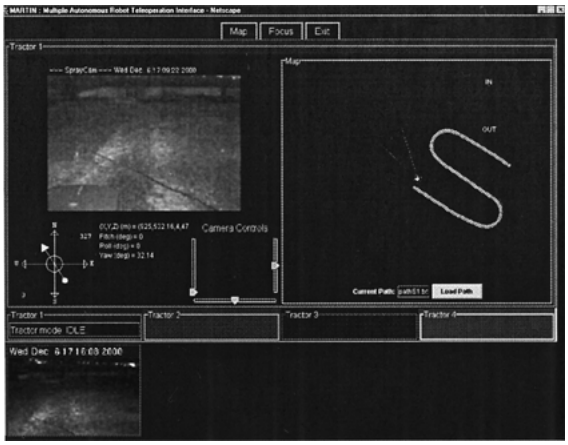

Figure 4. Remote operator interface. Operator interface showing focused view of a single tractor during playback, including video, status, and location on the map. A global map view and focused views for the other tractors are also available.

shows the focused view of the single operational tractor. It contains a larger video image, a compass display showing the vehicle heading, status information, and controls for the pan-tilt-zoom camera. The map on the right gives the location of the tractor, the previously taught path, and the actual path the tractor has driven autonomously so far. To give the operator context while using the pan-tilt-zoom camera, the camera's approximate field of view is drawn on the map. Using our wireless link, we also have the capability to remotely start, stop, and resume after an obstacle is detected.

4. Position-Based Navigation

In our application, we are concerned with the absolute repeatability of the tractor's position as it drives over a pre-taught path. Therefore, the path representation, position estimation, and navigation are all performed in a consistent world frame based on GPS. The control point of our vehicle is located at the center of its rear axle directly beneath the GPS antenna mounted on its roof. This point is near the hitch of the tractor, so towed implements follow a path similar to the results we show for the tractor. We currently do not have the capability to directly measure the position of a towed implement.

Position errors for a large vehicle such as a tractor can potentially cause damage to the vehicle as well as the surrounding environment. This dictates the use of a highly reliable position estimation and control system that is robust to sensor failures and can alert the operator if there is a problem that requires intervention. The following sections describe how the position estimation and path tracking algorithms used on our tractor achieve this goal.

4.1. Position Estimation

As described in the introduction, other researchers have automated agricultural vehicles. However, some of these systems rely on a single sensor such as camera-based visual tracking (Billingsley and Schoenfisch, 1999; Gerrish et al., 1997), or differential GPS (DGPS) (O'Conner et al., 1996). With only a single sensor used for localization, these systems must stop when a sensor failure occurs, or else the vehicle could cause damage because it does not know its correct position. Furthermore, without redundant sensors it becomes more difficult to detect when a sensor fails. Camera-based visual tracking can fail because of many problems, including changing lighting conditions, crop variation, and

adverse weather (Gerrish et al., 1997). DGPS also has problems because it relies on radio signals that can lose signal strength, become occluded, or suffer from multipath problems (Kerr, 1997; Neumann et al., 1996). We have experienced these problems with DGPS, especially while testing in a Florida orange grove that had trees taller than the antenna on our tractor.

To avoid the above problems, other researchers have used combinations of complementary sensors, such as DGPS, odometry, and visual tracking. These sensors have been combined using a voting strategy (Pilarski et al., 1999) or a rule based fusion module that looks at information availability from the different sensors (Zhang et al., 1999). While these methods can handle a sensor failure for a short period of time, they do not maintain a confidence estimate in their position estimate, and therefore cannot make decisions about when it is safe to continue. One solution to help ensure the vehicle's safety is to stop operation whenever it does not have a good DGPS signal (Pilarski et al., 1999), but this could result in extended downtime if the loss of DGPS signal is due to the geometry of the environment.

Our tractor uses a redundant set of sensors that complement each other. This redundancy allows the tractor to detect sensor failures and combine multiple measurements into a higher quality estimate. The primary navigation sensor is a Novatel RT2 dual frequency real-time kinematic carrier phase differential GPS receiver capable of better than 2 cm standard deviation absolute accuracy. This receiver uses an internal Kalman filter to output the uncertainty estimates on the measurements (Neumann et al., 1996). The tractor also has a KVH ECore 2000 fiber-optic gyroscope that precisely measures the heading rate changes of the vehicle (standard deviation 0.0001 rad/sec), custom wheel encoders that give a distance measurement that can be converted into a forward velocity (standard deviation 0.47 m/sec) and give a secondary measure of heading changes, and a doppler radar unit that is commonly used on farm equipment to measure forward speed (standard deviation 0.13 m/sec) even with wheel slip. These sensors are reliable and provide accurate differential information but suffer from drift over time. By combining the reliability of the encoders, gyro, and radar unit with the absolute reference of the DGPS system, the tractor can maintain an accurate position estimate that is robust to periodic DGPS dropout.

We use an Extended Kalman Filter (EKF) to combine the information from the different sensors described above into a single position estimate while also providing the uncertainty in that estimate. This uncertainty is used to make decisions about whether the vehicle can safely continue or if it needs assistance. Doing this allows the tractor to continue operating during a DGPS failure for as long as possible, given the position accuracy requirements of the particular application.

Under the assumptions of white Gaussian noise corrupted measurements and a linear system model, the Kalman filter provides the optimal estimate (Maybeck, 1982). The EKF is an extension of the Kalman filter to handle nonlinear system models by linearizing around the current state estimate. The EKF utilizes a model of the system to predict the next state of the vehicle. This allows the filter to compare sensor measurements with the expected state of the vehicle and reject sensor measurements that are not consistent based on a likelihood ratio test.

We chose to start with a simple 2D tractor model and it has given us sufficient performance. The model makes two important assumptions. It assumes that the vehicle has the non-holonomic constraint that it can only move forward, not sideways. It also makes a low-dynamics assumption that the forward and angular accelerations of the vehicle are essentially constant over a single time step. Making these assumptions explicit in the model allows the filter to reject measurements that do not follow these assumptions.

The state vector in our model includes global position x, y, forward velocity v, global heading θ, and heading rate $\dot{\theta}$. The non-linear system model used in the EKF that obeys the above constraints is given by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathtt{w} \tag{1}$$

where $\mathtt{w}$ is a vector of white Gaussian noise representing the uncertainty in the model. The measurements are given as linear combinations of the states as shown below

$$\begin{bmatrix} x_{gps} \\ y_{gps} \\ v_{radar} \\ v_{encoder} \\ \dot{\theta}_{encoder} \\ \dot{\theta}_{gyro} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v \\ \theta \\ \dot{\theta} \end{bmatrix} + \mathtt{v} \tag{2}$$

where v is a vector of white Gaussian noise representing the uncertainty in the measurements. As mentioned earlier, the DGPS receiver gives continuous estimates of the uncertainty in its measurements, so these values are used for the first two entries in v. The remaining entries are constants found from static tests of the other sensors. Equation (2) shows that the DGPS, radar, and gyro give direct measurements of state variables. The rear encoders on our vehicle measure wheel angular displacements $\Delta\alpha_L$ and $\Delta\alpha_R$ but can be converted into forward and angular yaw velocities using (Crowley, 1989)

$$v_{encoder} = \frac{R(\Delta\alpha_L + \Delta\alpha_R)}{2\Delta T} \tag{3}$$

$$\dot{\theta}_{encoder} = \frac{R(\Delta\alpha_L - \Delta\alpha_R)}{B\Delta T} \tag{4}$$

where R is the rear wheel radius, B is the track (distance between the rear wheels), and ΔT is the time interval. While these conversions can introduce noise into the measurements, they allow easy integration into the filter structure.

Using the noise estimates and models from Eqs. (1) and (2), the EKF prediction and correction equations (Maybeck, 1982) are run using whichever measurements are available at a given time step. This is possible because the measurement errors in v are assumed to be uncorrelated between sensors. At each time step, the filter outputs an estimate of the state of the system and its associated covariance.

Figures 5 and 6 show the behavior of our position estimator for three different types of sensor problems: outliers, dropout, and degradation. Figure 5 shows an overhead view of a path driven by the tractor. Various sensor problems were simulated during this run. These problems are similar to actual problems we have experienced with our DGPS unit, but simulating them allows a comparison of the filter output to the actual position of the tractor as given by the DGPS. The dashed line shows the DGPS baseline position measurement, while the solid line gives the output of the filter. The x's are the DGPS measurements that were presented to the filter. Every five meters along the path, an ellipse representing the 1-sigma uncertainty in the position measurement is plotted. Figure 6 shows the same information in a different format. The center line is the difference between the position estimate and the DGPS baseline as a function of distance along the path. The upper and lower symmetric lines give the 1-sigma uncertainty in the position estimate along the direction of maximum uncertainty.

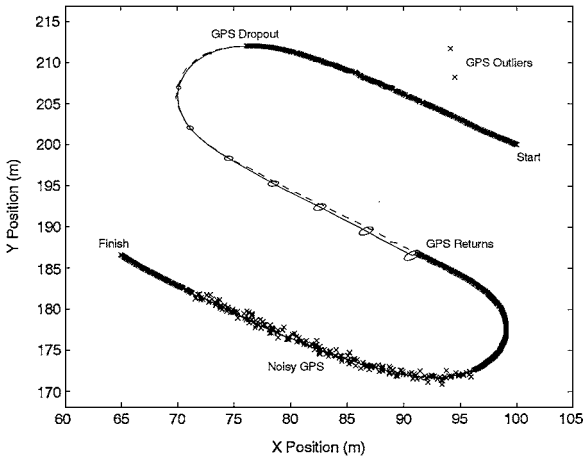

Figure 5. Position estimation. Overhead view of position estimate during outliers, dropout, and sensor degradation. The dashed path is the DGPS baseline, the x's are the measurements presented to the filter, the solid line is the filter output, and the ellipses give the 1-σ uncertainty in the estimate.

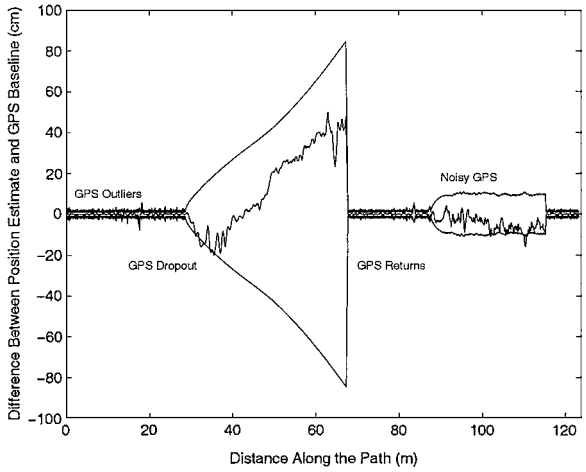

Figure 6. Position estimation error. Center line is the difference between the position estimate and the DGPS baseline. Surrounding lines give the 1-σ uncertainty in the position estimate along the direction of maximum uncertainty.

Near the beginning of the run, two incorrect DGPS measurements were given to the filter. However, since the filter uses its internal uncertainty estimate to test the probability of measurement validity, the filter was able to reject these measurements as outliers. Measurements are classified as outliers if they are beyond 3-sigma from the current estimate. Figure 6 shows that neither the estimate nor the uncertainty suffered from these false measurements. Other false measurements such

as excessive wheel spin causing misleading encoder measurements are similarly filtered out. Prolonged outliers have the same effect as sensor dropout, which is described next.

During the first turn, the filter stopped receiving DGPS measurements. The estimate was then based on dead-reckoning alone using the gyro, encoders, and radar. The uncertainty ellipses in Fig. 5 and the uncertainty lines in Fig. 6 show that the uncertainty of the estimate integrates with distance (the uncertainty does not grow linearly because of the curve in the path and the nonlinear filter update). The uncertainty ellipses become wider transverse to the path. This reflects the fact that a small error in the heading becomes large after it is integrated over a long distance. Using these uncertainty estimates, the tractor can find the probability that the vehicle has deviated beyond an application-specific threshold, and stop the tractor if necessary. The remote operator would be alerted to the problem and could correct the situation. However, because DGPS dropout is often caused by occlusion, the vehicle may be able to reacquire DGPS if it can keep moving to a new location. Figures 5 and 6 show this case. The tractor was able to run on dead-reckoning long enough for DGPS to return, at which point the estimate and uncertainty collapsed to their previous levels before DGPS dropped out.

After the second turn, the filter was given less precise DGPS measurements. Since our DGPS unit outputs estimates of the variance of its measurements, the filter was able to incorporate the noisy measurements correctly, and effectively low-pass filter them when it combined the DGPS measurements with measurements from the other sensors. Figure 5 shows that the position estimate remained smooth and close to the baseline despite the noisy measurements. As shown in Fig. 6, the noisy DGPS measurements caused an increase in the position uncertainty, but it remained bounded because of the availability of absolute (noisy) position measurements. Our DGPS receiver occasionally gives measurements with increased noise such as this when it has fewer satellites, it temporarily loses differential correction signals, or during multipath situations. We have observed standard deviation estimates up to 30 cm from the receiver, and our tests have shown that the actual measurements normally have a somewhat smaller standard deviation than the receiver's estimate of its uncertainty. The simulated measurements shown in Figures 5 and 6 have a standard deviation of 30 cm.

Because the filter was designed to handle sensor dropouts, our system degrades gracefully and will continue to perform well using only DGPS measurements to estimate the entire state vector. Without the high bandwidth of the gyro, the heading estimate will lag somewhat, but more importantly, a DGPS dropout in this scenario would cause the tractor to stop immediately because there would be no other measurements to provide backup information. If the dropout was caused by the geometry of the environment, and the tractor was forced to stop, there would be no chance of DGPS recovery and human intervention would be required.

This section illustrates two important points. First, using redundant sensors is advantageous because their measurements can be combined to form a better position estimate, and their measurements can be compared to determine if a sensor is malfunctioning. Second, using a filter that determines the uncertainty in its estimate allows the vehicle to continue driving autonomously as long as it has determined it is safe to do so. As described in Section 2.1, there is a trade-off between false positives and false negatives. By setting the thresholds conservatively, the filter is able to err on the side of false positives, thereby gaining safety at the expense of a few more interventions by the human operator to check whether the tractor is off course or in a dangerous situation.

Figures 5 and 6 show that despite the use of encoders, radar, and a gyro, the uncertainty in the position estimate grows when DGPS measurements are not available. This is because DGPS is the only absolute measurement that does not suffer from drift. Also, the majority of the uncertainty is due to errors in the vehicle heading that are integrated over time. We plan to improve this somewhat by incorporating data from the other sensors on our vehicle: front-wheel encoders, steering angle potentiometer, and a roll-pitch sensor (this would eliminate many of the spikes in Fig. 6 that were caused when the DGPS antenna on the roof of the vehicle swung more than 10 cm to the side as the vehicle rolled while driving through a rut) and experimenting with a more accurate vehicle model. However, because none of these measurements are absolute, they would still not bound the uncertainty to allow extended runs without DGPS. To help solve this problem, we plan to incorporate measurements of absolute heading from camera images of the crop rows themselves. These measurements will be incorporated into the filter using outlier detection so that errors in the vision system don't corrupt the state estimate. We will also look

into more advanced methods of fault detection and fault handling.

4.2. *Path Tracking*

Given the nonlinear nature of an Ackerman steered vehicle with an electrohydraulic steering system such as a tractor, the control problem is not trivial. However, simple controllers can give good results. One approach is to use a proportional controller on a point ahead of the vehicle (Billingsley and Schoenfisch, 1995; Gerrish et al., 1997; Pilarski et al., 1999; Wallace et al., 1985). Another approach is to linearize about the path, and then use a PI controller (Noguchi and Terao, 1997), a Linear Quadratic Regulator (O'Conner et al., 1996), or a Feedforward PID controller (Zhang et al., 1999). All of these approaches have given satisfactory performance for the relatively low speeds that farm vehicles normally travel. We have chosen to track a point ahead on the path because it is simple to implement and tune, and it has given good performance in a variety of circumstances.

Because of the variety of maneuvers that a tractor or other vehicle may need to execute, a general path tracker was developed that can follow arbitrary paths made up of points $[x, y, \theta]$ assuming that the spacing of the points is small relative to the length of the vehicle and the curvature of the path is never greater than the maximum curvature of the vehicle. The set of points that make up the path are stored in a position based hash table to allow fast retrieval of nearby points in large paths. The vehicle can start anywhere on the path as long as its position and orientation from the path are within thresholds set for the application. The tracker uses the position estimate from the Extended Kalman Filter described in the previous section. For safety, the tractor will not operate autonomously unless the position estimate uncertainty is within thresholds.

The inputs to the vehicle are desired speed and curvature, and our tractor has a low-level controller that sets and maintains these two quantities. We placed the origin of the tractor body frame at the center of the rear axle, as shown in Fig. 7. This has the effect of decoupling the steering and propulsion because vehicle curvature becomes determined by the steering angle ϕ alone (Shin et al., 1991). The desired speed of the vehicle can then be set by the application. The tracker takes the desired path and the current state of the vehicle and computes the curvature required to stay on the path. A simple kinematic model that gives the change in position $[x, y, \theta]$ of the vehicle for a given velocity v and curvature κ is

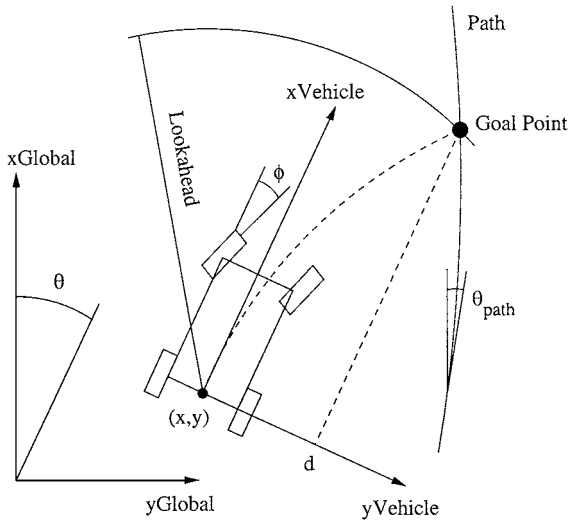

Figure 7. Pure pursuit. Diagram showing vehicle pursuing the goal point on the path one lookahead distance away.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ v\kappa \end{bmatrix} \tag{5}$$

The tracker uses a modified form of the Pure Pursuit algorithm (Amidi, 1990). The basic algorithm calculates a goal point on the path ahead of the vehicle, and the tracker then pursues this point, much like a human driver steers towards a point ahead on the road. This goal point is located a distance one lookahead l away from the vehicle. Figure 7 shows how the algorithm works. The tracker finds the closest path point and then walks up the path, interpolating to find the goal point one lookahead distance away. The goal point is then transformed into the vehicle's coordinate frame to find the y-offset d. It is shown (Amidi, 1990) that the circular arc that connects the vehicle to the goal point has a curvature given by

$$\kappa = \frac{2d}{l^2} \tag{6}$$

Equation (6) shows that Pure Pursuit is simply proportional control with the error signal computed one lookahead in front of the vehicle. The recorded path has heading information as well as position information. We have found empirically that for the type of paths that we are tracking, adding an error term between

the current vehicle heading θ and the recorded heading on the path θ_{path} both increases controller stability during path acquisition and reduces the error during path tracking. This change is given in the following equation for the curvature

$$\kappa = \frac{2d + K(\theta_{path} - \theta)}{l^2} \tag{7}$$

where K is a tunable constant.

The kinematic model Eq. (5) shows that the state variables integrate with distance traveled. This means that when the vehicle has a high forward velocity, a small change in curvature will result in large changes in $[x, y, \theta]$. The curvature Eqs. (6) and (7) show that increasing the lookahead distance l reduces the gain of the tracker. These observations suggest that when the vehicle has a higher forward velocity, a larger lookahead distance should be used. Making the lookahead a function of velocity provides a single path tracker that works for the entire range of operational speeds.

Despite the simplicity of the tractor model and the Pure Pursuit algorithm, the tracker has performed well over a variety of speeds and through rough terrain. Figure 8 shows a taught path recorded using the Teach interface. Overlaid on the desired path is the actual path that the tractor drove under control of the path tracker at a speed of 5 km/hour. The error profile for this run in Fig. 9 shows that the tracking error is the same for straight segments and curved segments. The results for an 8 km/hour run are similar but have errors of larger magnitude. The Pure Pursuit algorithm does not take

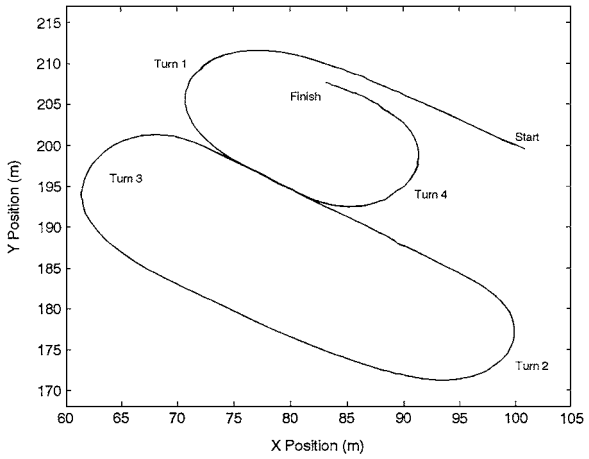

Figure 8. Path tracking. Overhead view showing a taught path and the autonomously driven path on top (the paths are only centimeters apart, so they appear indistinguishable).

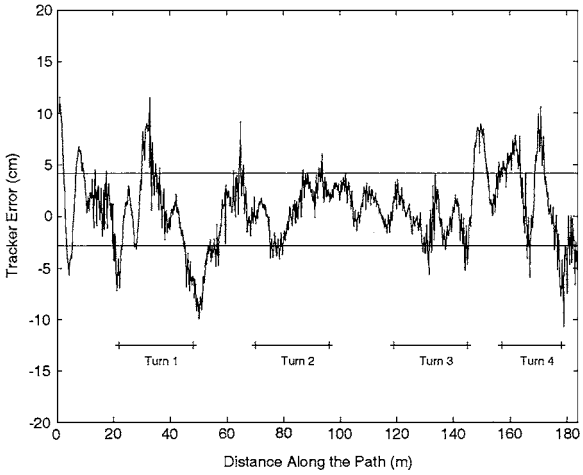

Figure 9. Path tracking error. Error between vehicle's position estimate and recorded path using DGPS baseline. The lines represent 1-σ bounds for the error.

vehicle dynamics into account, and assumes that steering changes on the vehicle occur instantaneously. As the speed of the vehicle increases, the vehicle dynamics play a greater role, and the error of this algorithm increases. However, for the operating speed ranges that we are interested in, the errors have all been within acceptable tolerances.

Table 1 gives a comparison of the errors at different speeds. The errors were calculated by comparing the vehicle's perceived position to the recorded path positions. Therefore, these numbers more accurately reflect the error of the tracking algorithm than the true error of the system. Since the overall repeatability of the entire system is really what matters, we performed a test to judge absolute repeatability. While teaching the path in Fig. 8, we used a spray paint can mounted on the bottom of the center rear axle of the tractor to mark a line on the ground. Then, while the tractor autonomously drove the path, a small camera mounted next to the spray can recorded video of the line as the tractor passed over it. From this video, we took 200 random samples for each speed and computed the actual error between the original path and the autonomously driven path. This gives an indication of the total error in the system because it includes errors in the position estimate as well as errors in the path tracker. This test was performed on a relatively flat grassy surface that contained bumps and tire ruts from repeated driving of the tractor.

The results of this analysis are shown in Table 2. The similarity between the results in Tables 1 and 2

Table 1. Path tracking error using DGPS baseline.

Speed	Bias (cm)	1-σ (cm)	Max (cm)
5 km/hr	0.67	3.50	11.58
8 km/hr	1.20	8.78	28.35

Table 2. Path tracking error using ground truth camera.

Speed	Bias (cm)	1-σ (cm)	Max (cm)
5 km/hr	1.47	3.18	7.62
8 km/hr	0.93	6.70	13.34

verify the repeatability of the position estimation system as well as the path tracking algorithm. The ground truth results actually have lower variance and maximum deviation values than the DGPS baseline results. This is because the path tracking algorithm acts to filter out erroneous DGPS readings caused by the antenna swinging to the side as the vehicle drives through ruts. These faulty readings degrade the recorded error using the DGPS baseline, but do not substantially affect the ground truth measurement because the camera and spray paint can were mounted near to the ground. Incorporation of a tilt sensor to correct for this would make the results more comparable.

While in a Florida orange grove, we tested the path tracker while pulling a 500 gallon air-blast sprayer. The grove was generally flat and grassy, but every other row between the trees of the grove was approximately 50 cm lower since these rows were used for irrigation. Since they were lower, these rows were also sometimes muddy. Tracking performance did not seem to change much despite the changing load as the water in the tank was sprayed out. We plan to test our system with other implements and in more challenging terrain in the future.

More advanced controllers that incorporate vehicle dynamics will also be investigated to allow the tractor to safely drive at higher speeds.

5. Obstacle Detection and Safeguarding

An automated tractor that operates on a farm needs to detect obstacles in order to: (1) provide safety for humans that get too close to the vehicle; (2) avoid causing damage to the environment (by collisions with trees, tools or other equipment located on the vehicle's path); and (3) avoid damaging or incapacitating itself due to ditches, irrigation canals, rocks, gopher holes, etc. These three factors make false negatives expensive; as a result, having a reliable and robust obstacle detection system is a hard prerequisite for any kind of automation in a real world setting.

Unfortunately, the fact that the natural environment contains very little structure and presents a large number of uncontrollable factors makes outdoor obstacle detection very difficult. The difficulty is proportional to the generality required from the system. While finding obstacles on very flat terrain is easily solved, creating a general system that will work in row crops, orange groves and other agricultural settings at any time of the day and in any weather can prove to be extremely complex.

Part of the problem is due to the fact that no sensor exists that can guarantee detection in the general case. Each sensor has failure modes that can make the automated vehicle unsafe unless there are other sensors that can cover these failure modes. For camera-based sensing, the changes in natural light that occur during the day, the limited dynamic range of the cameras and the various algorithmic limitations (like the lack of texture for stereo vision) significantly diminish obstacle detection capabilities. Another sensor commonly used for obstacle detection is a laser range finder, with either one- or two-axis scanning motion. But two-axis lasers can be very expensive and slow, if we take into consideration the requirements imposed by vehicle motion; one-axis lasers are cheaper and faster, but this advantage comes at the expense of significantly less data than provided by a two-axis laser.

Most work in agricultural automation has focused on the control aspects of automation and has ignored the obstacle detection problem (Billingsley and Schoenfisch, 1995; Gerrish et al., 1997; Noguchi and Terao, 1997; Zhang et al., 1999). The DEMETER system (Ollis and Stentz, 1996, 1997; Pilarski et al., 1999) detected obstacles by identifying objects that differ from a model of the cut and uncut crop. However, the performance of the obstacle detection system was not presented in detail, and the algorithms employed were relatively simple.

Obstacle detection systems have been developed for cross country navigation. The most common technique employed uses neural networks for terrain classification, operating on raw RGB data (Davis, 1995) or on features, such as intensity variance, directional texture,

height in the image, and statistics about each color band (Marra et al., 1988). The MAMMOTH system (Davis, 1995, 1996) drove an off-road vehicle. MAMMOTH used two neural networks (with one hidden layer each) to analyze separately visual and laser rangefinder data. Each neural network output a steering command. A "task neural network" took as inputs the outputs of the hidden layers in the two networks and fused the information into a steering command. The JPL system (Belluta et al., 2000) detected obstacles in an off-road environment by combining geometric information obtained by stereo triangulation with terrain classification based on color. The terrain classification was performed using a Bayes classifier that used mixtures of Gaussians to model class likelihoods. The parameters of the mixture model were estimated from training data through the EM algorithm. All of these systems have met with some success in addressing a very difficult problem. The main difference between our work and the prior work is that the agricultural application domain allows us to impose simplifying constraints on the problem to our advantage.

5.1. Technical Approach

At the high level, we have identified three ways to reduce the difficult problem of general obstacle detection to a more solvable one:

- Extract as many cues as possible from multi-modal obstacle detection sensors and use sensor fusion techniques to produce the most reliable result.
- Take advantage of the repeatability of the task by learning what to expect at a given location on the terrain.
- Since the system can call for human intervention, take a conservative approach to obstacle detection and allow a few false positive detections in order to drastically reduce the likelihood of false negative detections.

Our current sensor suite is shown in Fig. 10. It consists of two color CCD cameras used for obstacle detection and a Sony D30 pan-tilt color camera used for remote monitoring. The two Sony EVI-370DG cameras have a maximum resolution of 640 × 480 and a field of view that can vary from 48.8 to 4.3 degrees horizontally and 37 to 3 degrees vertically. We used a zoom setting that corresponds to approximately 30 degrees horizontally. The two color CCD cameras

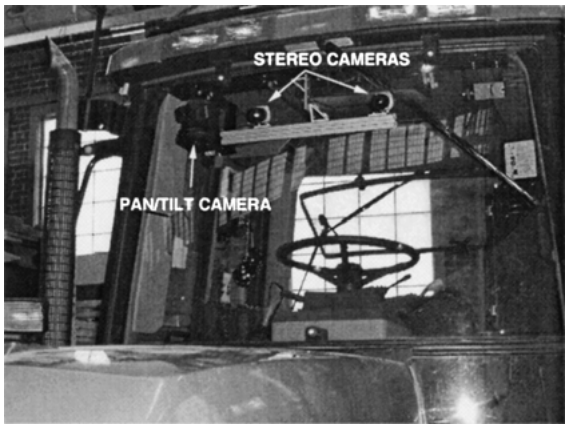

Figure 10. The sensor rack mounted inside the tractor cab. The two cameras used for stereo can slide sideways to vary the baseline. The pitch angle can be adjusted in order to change the look-ahead distance in front of the tractor. The pan-tilt color camera is used for the remote operator interface.

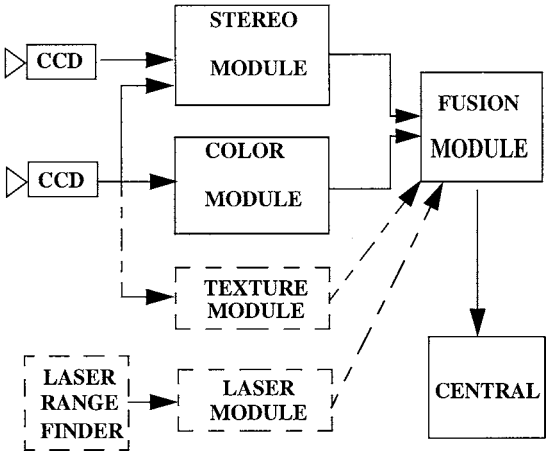

Figure 11. ODS architecture. Solid boxes correspond to modules for which we present results. Dotted ones represent modules that constitute future work.

provide color and geometric cues about the scene. In the future we will also use them to extract texture cues.

The current architecture of the obstacle detection system (ODS) is shown in Fig. 11. The ODS currently uses just the stereo range data and color data from the CCD cameras. The appearance and range data are separately classified and the results are integrated in the fusion module, which sends obstacle detection confidences to the vehicle controller (central module). The central module stops the tractor when the confidence exceeds a preset threshold.

The color module uses a three-layer artificial neural network trained with back propagation to segment

incoming images into classes defined by the operator during the training process. In the orange grove scenario, such classes are "grass", "tree", "sky", "obstacle", etc. The "obstacle" class corresponds to colors that are not characteristic of any other class. A human operator provides the training data by manually classifying portions of recorded images. This training can be accomplished in minutes due to a graphical user interface that simplifies the process. In the future we are planning to use unsupervised clustering techniques that will expedite the training process even more.

At run time, the 320×240 pixel images are split into patches of 4×4 pixels, and each pixel is represented in the HSI color space. The intensity component is discarded for less sensitivity to brightness, and the hue is expressed by two numbers (sin(H) and cos(H)) to eliminate discontinuities. Thus, the input layer of the neural network has $4 \times 4 \times 3$ inputs. We selected an input patch of 4×4 pixels because it is good compromise between a small patch, which is sensitive to noise, and a large one, which permits only a coarse-scale classification.

The output layer contains as many output units as classes used in the training data. In order to label an input patch as belonging to class C, the output unit corresponding to class C should have a value close to "ON" and the other units should have values close to "OFF". If a patch does not fit well into any other class, it is labeled as an obstacle patch. Thus, even though an obstacle class is defined for training purposes, the system can still correctly classify obstacles that were not represented in the training set.

In our test scenario we only used the color segmentation for obstacle detection, so we defined two classes (obstacle/non-obstacle) in order to avoid unnecessary complexity in the neural network's decision surface.

To determine the optimal number of units in the hidden layer, we started with a large number (allowing the network to overfit the data) and gradually reduced them until the total classification error on a standard test began to rise. We arrived at three hidden units, which was the smallest number that did not degrade the performance of the network on the test set.

The stereo module is currently based on a 3-D "safety box" representing a navigation zone in front of the tractor. The module creates a disparity map and signals the presence and the location of any objects within this box. While this is certainly a minimalistic approach to obstacle detection with stereo vision, the module provides very fast and reliable results for obstacles that protrude significantly upward from the terrain.

For the fusion module, we have the choice of performing sensor-level or central-level fusion. The sensor-level fusion consists of computing a measure of confidence at each separate sensor, and then summing over all the individual sensors to obtain the global confidence. In central-level fusion, the sensor data is minimally processed before combination and the computed confidence is a more complex function of the multi-modal data.

In general, data can be used more efficiently through central-level fusion than through the sensor-level paradigm (Klein, 1993). However, this approach has several disadvantages: (1) it does not distribute the computational load, (2) it requires precise calibration of the sensors and registration of the data; and (3) it is difficult to incrementally add new sensors. In our approach, we opted to investigate sensor modalities one at a time, understand their individual contributions, and fuse their data at the sensor level using simple strategies. Once we have identified the best sensors and data modalities, we will explore more sophisticated fusion strategies at the central level to improve the performance of the system.

5.2. Experimental Results

We performed two kinds of experiments, meant to test two of our claims: (1) that redundancy in sensors and the use of multi-modal data reduce overall failure modes; and (2) that increasing the locality of the training data improves the overall performance.

5.2.1. Multi-Modal Data. To test the effectiveness of using multi-modal data, we measured system performance on data that represented failure modes for the individual modality types: appearance (color) and range (stereo).

For the first test, we used the scene depicted in Fig. 12, which shows a human in a green suit against a green background. Since the neural network was trained to consider the green grass a non-obstacle, the response of the color module to the human obstacle was not very strong, as shown in Fig. 13. However, since the stereo module uses only geometric information, it is unaffected by the similarity in color between the obstacle and the background, and it gives a strong response indicating the presence of an obstacle. As a result, even a simple fusion strategy consisting of the

Figure 12. Failure mode for color-based ODS: A human with a green suit against green background.

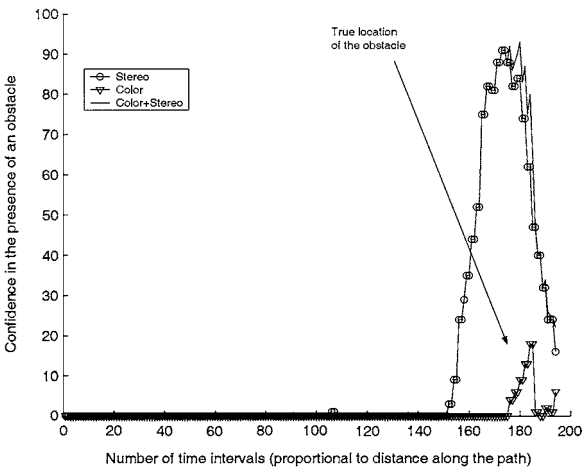

Figure 13. ODS responses for the obstacle in Fig. 12, as the tractor approaches the obstacle. The small peak produced by the color module arises from the skin color; when the human is very close, the area in the image occupied by skin becomes large enough to exceed the obstacle threshold.

sum of the responses of the two modules (color and stereo) results in reliable detection.

The symmetric case is depicted in Fig. 14: a tarp placed in the path of the tractor is invisible to the stereo module. However, as shown in Fig. 15, the color module is able to detect it, and the ODS reports an obstacle with high confidence.

We have presented the extreme cases here, but the benefit of multiple sensors and data modalities becomes even more significant when class differences are subtle. When the individual response from each sensor is not much higher than the noise level, the correlation

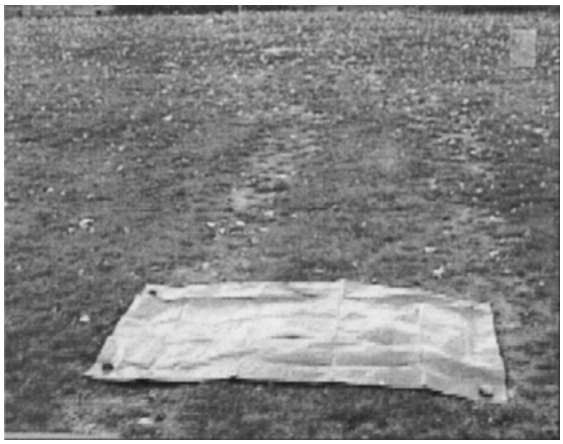

Figure 14. Failure mode for stereo-based ODS: A tarp on the ground.

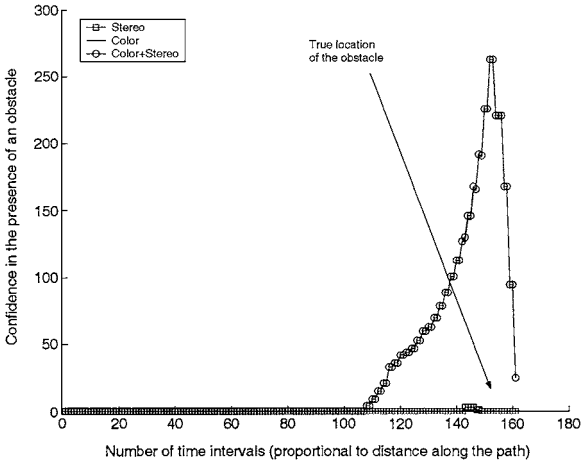

Figure 15. ODS responses for the obstacle in Fig. 14, as the tractor approaches the obstacle.

between these responses can still produce a cumulative effect that trips the threshold and stops the vehicle.

More recent experiments in natural scenes (see Fig. 16) confirm the fact that combining different cues for obstacle detection results in more reliable obstacle detection.

5.2.2. Data Locality. Agricultural operations are very repetitive. In general, a vehicle drives a given route many times; therefore, it is possible to learn the appearance of terrain at a given location (and time) and use this information for obstacle detection purposes when driving autonomously. This amounts to obtaining training information from all the parts of the route that have different appearances. In our case, for the stereo module,

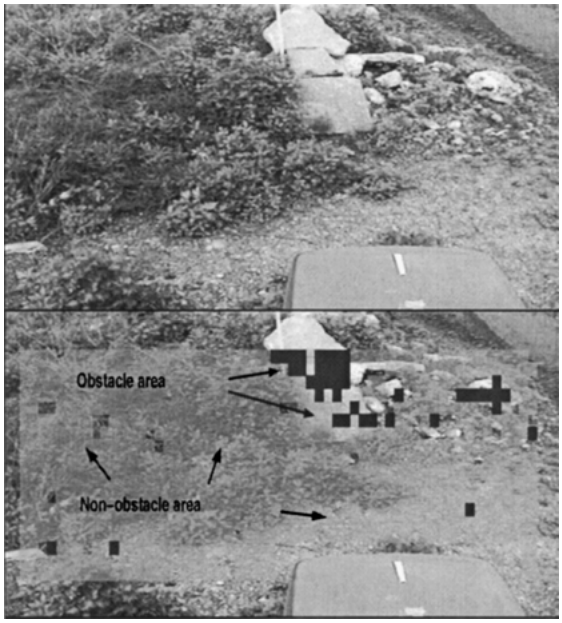

Figure 16. Results of a more recent version of our ODS system on a natural scene. Top: the raw image presenting rocks in front of the vehicle. Bottom: the same image, segmented using color and texture features. The light-gray area represents the obstacle-free region, the dark gray represents obstacles.

this could consist of storing the average height of the grass for each row in the orange grove, with the result that the "safety box" defined by the module is adjusted automatically depending on the location. For the color module, information about the current time of the day could be used in order to perform more effective color constancy or to use different versions of the neural network trained with labeled data collected at similar times of the day.

However, two factors affect the appearance of the terrain at a given location and time of day. First, ambient light and weather conditions vary in both predictable and unpredictable ways. To some extent, this effect can be cancelled using color constancy techniques. We used a gray calibration target mounted on the tractor in a place visible to the cameras. As the tractor drove, we repeatedly measured the color shift due to ambient light and corrected the input images accordingly. This approach did not solve the problem completely but greatly improved the color segmentation.

Second, the terrain at each location changes slowly over the course of a season. The crop, grass, and weeds grow. The plants change in appearance as well. These effects preclude a purely rote learning strategy (based on location and time of day) and require some generalization. We envision an obstacle detection system which learns not only how to classify terrain but how to predict changes over time. For example, the growing rate of the grass and the appearance changes of the vegetation could be learned and predicted.

We tested the value of data locality using data recorded in Florida. We performed the following experiment: we created two training sets containing images collected around two locations, A and B. We trained two versions of the color module's neural network, one on the data collected at location A, and one on the joint training set, containing the images from both locations. A typical image with an obstacle is presented in Fig. 17.

We then tested the two neural networks on a video sequence of the tractor approaching an obstacle. The sequence was recorded around location A. The responses of the two neural networks as the tractor approached the obstacle are presented in Fig. 18. As we expected,

Figure 17. Image from Florida containing an obstacle on the path of the tractor.

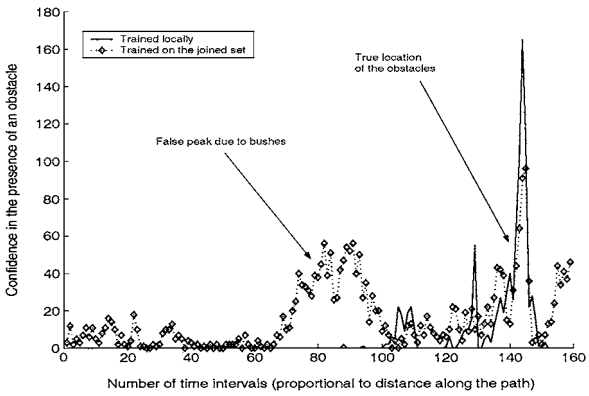

Figure 18. Confidence levels for the color-based ODS at location A. The solid line corresponds to the NN trained locally, and the dotted one to the NN trained on the joint set.

the neural network that was trained locally performed better. In the case presented, the network trained on the joint set generated a false obstacle peak due to some bushes that were correctly classified as vegetation by the locally trained network. Other experiments in Florida and Pittsburgh confirmed the intuition that the quality of the color segmentation is related to the locality of the training data.

The importance of local training data means that in the future we should focus on methods that are appropriate for recording the location and time of the day. Neural networks might not be the best learning algorithm for this, since the amount of training data required would be extremely large. We believe that memory-based learning together with automatic clustering techniques could be a better approach for our problem. In the memory-based paradigm, the system would periodically record data and perform automated clustering of the color data obtained from each image (or from images within some neighborhood). We would then model the data distribution at each location with a Gaussian mixture model. At runtime, we would use any classification algorithm to perform image segmentation, based on the distance from the clusters described by the Gaussian mixture model.

Other areas of future work include: (1) using information coming from a laser range finder as an additional cue; and (2) switching from the current sensor-level fusion paradigm to central-level fusion.

6. Conclusions

In addition to the component results for position estimation, path tracking, obstacle detection, and human intervention, the tractor was tested as a system in an orange grove in Florida. Figure 19 shows a plan view of the total path taken during autonomous runs in Florida. Overall, the system performed quite well, navigating accurately up and down the rows, even in the presence of hazards such as irrigation ditches. With commercial GPS units steadily improving in accuracy and dropping in cost, position-based navigation will become the method of choice for agricultural operations. The remaining problems include navigating in difficult terrain, such as on slopes, over bumps, and in mud; and navigating during dropout, including in areas with GPS occlusion.

But the real challenge for agricultural operations is safeguarding the people, the environment, and the machines. Until this problem is solved, we cannot field unmanned vehicles. The outlook is promising, however. By developing cautious systems, involving humans remotely, and capitalizing on the repetitive nature of the task, we can make substantial headway in solving a problem that is, in the general case, very difficult.

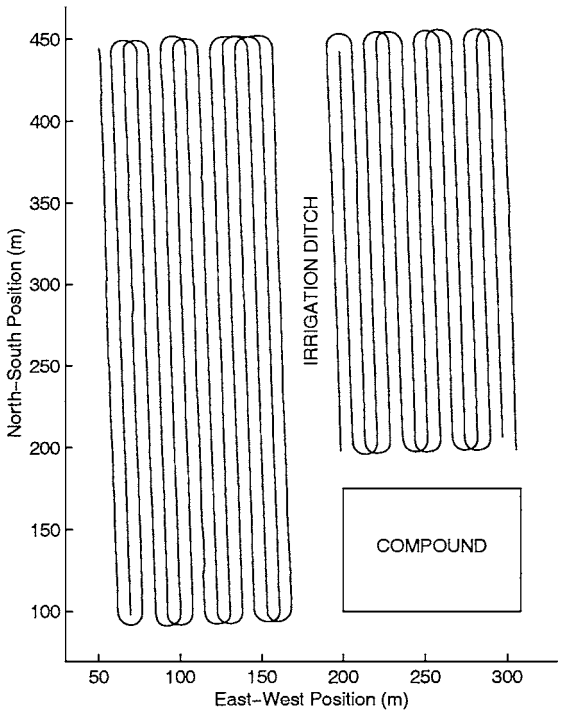

Figure 19. Overhead view of paths used during autonomous testing in a Florida orange grove. The turning radius of the tractor-sprayer combination requires the machine to travel down every other row.

Acknowledgments

The authors would like to thank the entire Autonomous Spraying Project group, past and present, including Chris Fromme, Frank Campagne, Jim Ketterer, Bob McCall, and Mark Ollis. Support was provided by NASA under grant number NCC5-223 and by Deere & Company via a tractor donation.

References

Amidi, O. 1990. Integrated mobile robot control. Masters Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA.

Belluta, P., Manduchi, R., Matthies, L., Owens, K., and Rankin, A. 2000. Terrain perception for DEMO III. *Intelligent Vehicle Symposium*.

Billingsley, J. and Schoenfisch, M. 1995. Vision guidance of agricultural vehicles. *Autonomous Robots*, 2.

Crowley, J.L. 1989. Control of translation and rotation in a robot vehicle. In *Proceedings of the IEEE Conference on Robotics and Automation*.

Davis, I.L. 1995. Neural networks for real-time terrain typing. The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU-RI-TR-95-06.

Davis, I.L. 1995. Sensor fusion for autonomous outdoor navigation using neural networks. The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU-RI-TR-95-05.

Davis, I.L. 1996. A modular neural network approach to autonomous navigation. The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU-RI-TR-96-35.

Erbach, D.C., Choi, C.H., and Noh, K. 1991. Automated guidance for agricultural tractors. *Automated Agriculture for the 21st Century, ASAE*.

Gerrish, J.B., Fehr, B.W., Van Ee, G.R., and Welch, D.P. 1997. Self-steering tractor guided by computer vision. *Applied Engineering in Agriculture*, 13(5).

Kerr, T.H. 1997. A critical perspective on some aspects of GPS development and use. In *16th Digital Avionics Systems Conference*, AIAA/IEEE, Vol. 2.

Klein, L.A. 1993. *Sensor and Data Fusion Concepts and Applications*. SPIE Optical Engineering Press: Bellingham, WA.

Marra, M., Dunlay, R.T., and Mathis, D. 1988. Terrain classification using texture for the ALV. In *Mobile Robots III*, *SPIE*, Vol. 1007.

Neumann, J.B., Manz, A., Ford, T.J., and Mulyk, O. 1996. Test results from a new 2 cm real time kinematic GPS positioning system. In *Proc. of ION GPS-96*, Kansas City.

Noguchi, N.K. and Terao, H. 1997. Development of an agricultural mobile robot using a geomagnetic direction sensor and image sensors. *Journal of Agricultural Engineering Research*, 67.

O'Connor, M., Bell, T., Elkaim, G., and Parkinson, B. 1996. Automatic steering of farm vehicles using GPS. In *Proc. of the 3rd Int. Conf. on Precision Agriculture*.

Ollis, M. and Stentz, A. 1996. First results in vision-based crop line tracking. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*.

Ollis, M. and Stentz, A. 1997. Vision-based perception for an autonomous harvester. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robotic Systems*.

Maybeck, P. 1982. *Stochastic Models, Estimation, and Control*, Academic Press: New York.

Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K., and Stentz, A. 1999. The DEMETER system for autonomous harvesting. In *Proc. of the 8th Int. Topical Meeting on Robotics and Remote Systems*.

Shin, D.H., Singh, S., and Shi, W. 1991. A partitioned control scheme for mobil robot path tracking. In *Proc. IEEE Int. Conf. on Systems Engineering*.

Southall, B., Hague, T., Marchant, J.A., and Buxton, B.F. 1999. Vision-aided outdoor navigation of an autonomous horticultural vehicle. In *Proc. 1st. Int. Conf. on Vision Systems*.

Tillet, N.D. 1991. Automatic guidance for agricultural field machines: A review. *Journal of Agricultural Engineering Research*, 50.

Wallace, R., Stentz, A., Thorpe, C., Moravec, H., Whittaker, W., and Kanade, T. 1985. First results in robot road-following. In *Proc. of the Int. Joint Conf. on Artif. Intell*.

Will, J., Stombaugh, T.S., Benson, E., Noguchi, N., and Reid, J.F. 1998. Development of a flexible platform for agricultural automatic guidance research. *ASAE Paper* 983202.

Yukumoto, O. and Matsuo, Y. 1995. Research on autonomous land vehicle for agriculture. In *Proc. of the Int. Sym. on Automation and Robotics in Bio-production and Processing*, Vol. 1.

Zhang, Q., Reid, J.F., and Noguchi, N. 1999. Agricultural vehicle navigation using multiple guidance sensors. In *Proc. of the Int. Conf. on Field and Service Robotics*.

Anthony Stentz is a Senior Research Scientist at the Robotics Institute, Carnegie Mellon University. His research expertise includes mobile robots, path planning, computer vision, system architecture, and artificial intelligence in the context of fieldworthy robotic systems.

Dr. Stentz has led a number of successful research efforts to automate navigation and coal cutting operations for a continuous mining machine, cross-country navigation for an unmanned ground vehicle, nuclear inspection tasks for a manipulator arm, autonomous landing for an unmanned air vehicle, crop row tracking for an autonomous harvesting machine, and digging operations for a robotic excavator.

Dr. Stentz received his Ph.D. in computer science from Carnegie-Mellon University in 1989. He received his M.S. in computer science from CMU in 1984 and his B.S. in physics from Xavier University of Ohio in 1982.

Cristian S. Dima is pursuing a Ph.D. degree in robotics at the Robotics Institute, Carnegie Mellon University. He received his B.A. in mathematics and computer science from Middlebury College in 1999.

His research focuses on computer vision and multisensor fusion applied to obstacle detection for autonomous outdoor robots.

Carl Wellington is pursuing a Ph.D. in robotics at the Robotics Institute, Carnegie Mellon University. He received his B.S. in engineering from Swarthmore College in 1999.

His research interests are in position estimation and control for outdoor autonomous robots. Currently he is focusing on the use of learning algorithms for the safe control of off-road vehicles.

Herman Herman is a Commercialization Specialist at the National Robotics Engineering Consortium. He has worked in the field of robotics and computer vision for the last 12 years. His present and past works include the automation of an underground mining vehicle, a real-time tracking system, a 6-D optical input device and 3-D image processing. His area of expertise includes image processing, automation and system engineering.

Dr. Herman received his Ph.D. in Robotics from Carnegie Mellon University in 1996 and B.S. in Computer Science from University of Illinois at Urbana Champaign in 1989.

David Stager is a Design Engineer at the National Robotics Engineering Consortium of Carnegie Mellon University. He received his B.S. in Electrical & Computer Engineering and Math & Computer Science from Carnegie Mellon University in 1995.

While at the NREC, he has helped design and build several fully autonomous robotic vehicles including forklifts, tuggers, a tractor, and various ATV's for commercial and military use.