is not the case with, for example, a neural net representation. The large number of free parameters allows the histogram to represent a wider range of density shapes than is possible with a multi-Gaussian PDF. Computing probability densities from the model is computationally fast when compared to any of the alternatives (particularly to K-nearest neighbor); a single table lookup followed by a single division operation produces the required result.

Note that the issues concerning the use of full RGB color versus normalized color are quite different for the obstacle detector than for shadow compensation and crop line detection. Normalized color is not used as an attempt to remove shadow noise; that is accomplished using the method described in Section 3. Here, normalized color is used to compensate for different iris openings between the training image and the images to be processed, and also as a means for reducing the dimensionality of the PDF space. While this does prevent the use of intensity as a metric for distinguishing obstacles, we have found that obstacles typically differ from crop significantly enough in color alone that the use of intensity information is not as necessary; such is often not the case for the crop line follower, which often must distinguish between the quite similar appearance of cut and uncut crop.

In order to perform robustly, it is likely that further development of the obstacle detector will be necessary; for example, the histogram PDF may need to be allowed to evolve over time to compensate for changing crop appearance. Field testing of the obstacle detector is still in a preliminary stage, though results such as shown in Figure 5 are promising.

6. Conclusions

Several different vision-based behaviors have been implemented for the Demeter automated harvester, and have been demonstrated successfully in real world conditions. A crop line tracking behavior, which adapts to local changes in the environment, has been successfully used to cut over 60 acres of alfalfa hay. Explicitly modeling and removing shadows in the outdoor environment, though a difficult problem in general, has proven partially amenable to approximation methods. Other behaviors, such as end-of-row detection and obstacle detection, show promising initial results.

These combined results demonstrate the feasibility of vision-based guidance in an agricultural environment. Such a system, when combined with the positioning capability allowed by GPS, appears viable for near term commercial development as either a driver aid or as a completely autonomous system.

Acknowledgments

The authors would like to acknowledge Mike Blackwell, Kerien Fitzpatrick, Mike Happold, Regis Hoffman, Alex Lozupone, Ed Mutschler, Henning Pangels, Simon Peffers, and Red Whittaker for their work on the Demeter harvester. This work was jointly supported by New Holland and NASA under contract number NAGW-3903.

References

[1] Billinglsey, J. and Schoenfisch, M. Vision-Guidance of Agricultural Vehicles. Autonomous Robots, Vol. 2, No. 1, 1995, pp. 65-76

[2] Brandon, J. Robert and Searcy, Stephen W. Vision Assisted Tractor Guidance for Agricultural Vehicles. Transactions of the Society of Automotive Engineers, 1993, paper 921650.

[3] Duda, Richard and Hart, Peter. Pattern Classification and Scene Analysis. 1973, J. Wiley & Sons, pp. 114-118.

[4] Gerrish, John et. al. Path-finding by Image Processing in Agricultural Field Operations. Transactions of the Society of Automotive Engineers, 1987, paper 861455.

[5] Healey, Glenn. Segmenting Images Using Normalized Color. IEEE Transactions on Systems, Man and Cybernetics, 1992.

[6] Klassen, N.D. et al. Guidance Systems for Agricultural Vehicles. Dept. of Mechanical Engineering, University of Saskatchewan, Saskatchewan, Canada.

[7] Novak, C.L. and Shafer, S.A. Color Vision. Encyclopedia of Artificial Intelligence, 1992, J. Wiley and Sons, pp 192-202.

[8] Ollis, Mark & Stentz, Anthony. First Results in Crop Line Tracking. Proceedings of IEEE Conference on Robotics and Automation (ICRA ‘96), Minneapolis, MN April1996, pp.951-956.

[9] Pomerleau, Dean. RALPH: Rapidly Adapting Lateral Position Handler. Proceedings of the 1995 IEEE Symposium on Intelligent Vehicles, Detroit, Michigan.

[10] Reid, J.F. and Searcy, S.W. An Algorithm for Separating Guidance Information from Row Crop Images. Transactions of the ASAE. Nov/Dec 1988 v 31 (6) pp. 1624-1632.

[11] Tsai, Roger. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pp. 323-344.

3) For each scan line i in the image, compute a score for the scan line S(i) as follows:

a) set S(i) = 0;

b) Increment S(i) for every scan line x from 0 (the top of the image) to i-1 for which F(x) = "afterEnd".

c) Increment S(i) for every scan line y from i+1 to i_MAX (the bottom of the image) for which F(y) = "beforeEnd".

4) The end of row is the row i with the highest score S(i).

We experimented with a broad range of binary evaluation functions F(i). Our current version first computes the crop line fitting algorithm described in Section 2 to each scan line. The location and height of the resulting best fit step functions are then compared to precomputed ranges gathered from training data; if they fall within the allowed ranges, the boundary is accepted as a genuine crop line and the row receives a "beforeEnd" label; otherwise, the row is labeled "afterEnd".

Our most common use for the end-of-row detection module is to trigger a transition into a turn behavior when the end of row is reached, as described in Section 1. In order to prevent a single spurious image from falsely causing an end-of-row trigger, this message is sent only after both 1) the distance to the end of row falls below some threshold and 2) a series of processed images indicate the end of row has been coming progressively nearer.

Although we have not yet gathered enough data to obtain accurate reliability figures, the system has successfully located several end-of-row points in field experiments. One problem with trying to ascertain the reliability of the end-of-row detector is the wide variety of situations which can be encountered; while the crop row tracker need only distinguish between cut and uncut crop, the end-of-row detector must be capable of dealing with images containing almost anything, such as the road and cow pasture which appear near the top of Figure 7.

5. Obstacle detection

The obstacle detection algorithm is used to locate potential obstacles in the camera's field of view. The method uses a training image to build a probability density function (PDF) for combined cut and uncut crop as a function of RGB pixel value. For each new image, shadows are compensated for as described in Section 3. Next, image pixels are marked whose probability of belonging to the crop PDF falls below some threshold. Finally, regions of the image containing a large number of such marked pixels are identified as obstacles. Figure 8 shows an example of such an image before and after processing; potential obstacles are marked as a solid region.

Traditionally, a wide range of representations have been used for PDFs; multi-dimensional Gaussian models,

Figure 8: Detecting potential obstacles.

K-nearest-neighbor approximations, histograms, and neural nets, just to name a few. These representations vary in computational efficiency and in the kinds of PDFs that can be represented. Each of these representations has advantages and disadvantages in training time, representational power, lookup time, and storage space. For this application, we used a discretized 2D histogram in normalized color space, with each cell containing an independent probability density estimate; the reasons for this choice are discussed below.

The discretization used is 12 bits; 6 for R/(R+G+B), and 6 for G/(R+G+B). Producing the histograms requires an independent estimation of probability for each of 64x64 = 4096 different discretized bins; it thus requires over 4000 parameters to describe. Compared to a multi-Gaussian representation, this may seem excessive; since there are so many free parameters, a large number of training samples are required in order to form a reasonable PDF. In our application, however, training data is plentiful, since every image pixel represents a training point. Further, updating the PDF with a new training point is quite rapid, since it simply requires incrementing a single counter; this

significant shadow; for our application values of C_{red} = 5.6, C_{green} = 4.0, and C_{blue} = 2.8 were found to work well.

An attempt was made to calculate C_{red}, C_{blue} and C_{green} values a priori from blackbody spectral distribution models of sunlight and skylight. This calculation produced qualitatively the correct result, e.g. $C_{red} > C_{green} > C_{blue}$; however, the a priori calculated values were found to be less useful than the experimentally determined values. This discrepancy may be due to a number of sources, such as the inadequacy of the blackbody spectral distribution function as a model for skylight and the variable sensitivity of the camera CCD to red, green, and blue light.

The method described above necessarily makes a number of simplifications. The red, green, and blue filters actually each pass a range of frequencies; SPD variations in sunlight and skylight within a single band are not taken into account. Shadowed areas can receive significant illumination from reflected light from neighboring sunlit areas; such interreflections are not modeled. The differing effects of lighting angle for sunlight and skylight are ignored, as are non-linearities in the CCD chip response.

The values of C_{red}, C_{blue}, and C_{green} depend on color of both the sunlight and the skylight. These colors can vary across different times of day and different atmospheric conditions. In our application, shadows typically cause the most trouble on cloudless days in the late afternoon; we therefore chose coefficients optimized for this case.

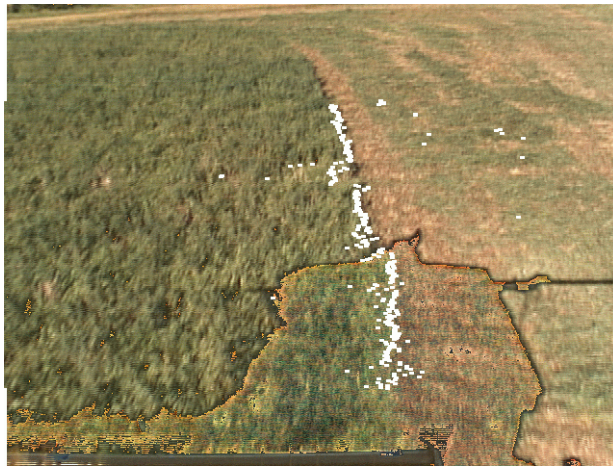

Figure 6: A successful example of shadow compensation.

Despite these limitations, applying this method allows crop lines to be successfully extracted from a number of images which would otherwise return incorrect results. For example, applying the shadow compensation method described above to the image shown in Figure 5 produces a much improved estimate of the crop boundary, as shown in Figure 6. In at least one field test, the shadow compensation allowed the harvester to successfully follow a crop line in an area in which it failed without the compensation. Further, the same set of constants C_{red}, C_{blue}, and C_{green} were found to work in the two locations for which shadowed images were collected (Kansas and Pennsylvania).

From initial testing, it appears that the method works better away from shadow edges; as can be seen in Figure 6, the compensation becomes increasingly inaccurate near shadow/sunlit boundaries, possibly because the simple two-source spectral distribution model breaks down.

4. Detecting the End of a Crop Row

The goal of the end of row detector is to estimate the distance of the harvester from the end of the crop row. When the end of row boundary is approximately perpendicular to the crop line, and the camera is mounted with zero roll (as in our system), the distance to the end of row is purely a function of the image row where the crop line boundary stops. Figure 7 shows an image which has been correctly processed; the white line marks the computed image row corresponding to the crop row end.

Figure 7: Locating the end of a crop row.

Our end of row detection algorithm attempts to find the image row i which most cleanly separates those scan lines containing a crop line boundary from those which do not contain such a boundary. The algorithm, described below, first uses a binary function F(i) to classify each image row according to whether it contains a crop row boundary; next, it searches for the row which best divides the “beforeEnd” rows from the “afterEnd”.

1) Digitize an image.

2) Remove shadow noise as described in section 3.

2) For each scan line i in the image:

Apply a binary evaluation function F(i) to determine whether row i contains a genuine crop line boundary. Let F(i) = “beforeEnd” if i contains a genuine boundary point, and F(i) = “afterEnd” if not.

Figure 5: Shadow noise.

rapidly, the prevalence and effect of shadow noise can vary dramatically on time scales of less than a second.

Normalizing for intensity, though an intuitively appealing method of dealing with shadow noise, fails to be useful in our application for two reasons. The primary problem is that it does not take into account the significant color changes present in shadowed areas. For example, normalizing the image in Figure 5 before processing still results in an incorrect crop line boundary estimate. A number of factors contribute to this color shift, but perhaps the most significant is the difference in illumination sources between the shadowed and unshadowed regions[5]; the dominant illumination source for the unshadowed areas is sunlight, while the dominant illumination source for the shadowed areas is skylight. A secondary problem with intensity normalization is that it prevents the crop line tracking algorithm from using natural intensity differences to discriminate between cut and uncut crop; depending on local conditions, such natural intensity differences can be a useful feature.

We present a technique for modeling and removing shadow noise which is based on compensating for the difference in the spectral power distribution (SPD) between the light illuminating the shadowed and unshadowed regions. In an ideal camera, the RGB pixel values at a given image point are a function of $S(\lambda)$, the spectral power distribution (SPD) emitted by a point in the environment [7]; for example, R is determined in Equation (1), where r_0 is a scaling factor and $\bar{r}(\lambda)$ is the function describing the response of the CCD chip and red filter; typically, this function falls to 0 outside of a narrow wavelength band.

$$R = r_0 \int S(\lambda)\bar{r}(\lambda)d\lambda \tag{1}$$

r_0 and $\bar{r}(\lambda)$ are purely functions of the CCD camera; our goal, therefore, is to construct a model of how shadows alter the function $S(\lambda)$.

To a first approximation, $S(\lambda)$ is simply the product of the SPD of the illuminating light, $I(\lambda)$, with the reflectance function of the illuminated surface point, $\rho(\lambda)$:

$$S(\lambda) = I(\lambda)\rho(\lambda) \tag{2}$$

Suppose we assume that every point in the environment is illuminated by one of two SPDs; either $I_{sun}(\lambda)$, comprising both sunlight and skylight, or $I_{shadow}(\lambda)$, comprising skylight only. Then the red pixel values for unshadowed regions will be computed by

$$R_{sun} = r_0 \int I_{sun}(\lambda)\rho(\lambda)\bar{r}(\lambda)d\lambda \tag{3}$$

and the red pixel vales for shadowed regions by

$$R_{shadow} = r_0 \int I_{shadow}(\lambda)\rho(\lambda)\bar{r}(\lambda)d\lambda \tag{4}$$

From Equations (3) and (4), we see that it is in general not possible to compute R_{sun} from R_{shadow} without knowledge of the reflectance function of the environment patch being imaged. This is problematic, because for our application, this reflectance function is always unknown. However, if we approximate $\bar{r}(\lambda)$ as a delta function with a non-zero value only at λ_{red}, then (3) and (4) simplify to

$$R_{sun} = r_0 I_{sun}(\lambda_{red})\rho(\lambda_{red}) \tag{5}$$

and

$$R_{shadow} = r_0 I_{shadow}(\lambda_{red})\rho(\lambda_{red}) \tag{6}$$

so that R_{sun} and R_{shadow} can be related by a constant factor C_{red}:

$$R_{sun} = R_{shadow}\left(\frac{I_{sun}(\lambda_{red})}{I_{shadow}(\lambda_{red})}\right) = R_{shadow}C_{red} \tag{7}$$

The same analysis can be repeated for the G and B pixel values. Under the assumptions given above, the parameters C_{red}, C_{blue}, and C_{green} remain constant across all reflectance functions $\rho(\lambda)$ for a given camera in a given lighting environment.

Implementing this shadow compensation therefore requires

1) the selection of appropriate constants for C_{red}, C_{blue} and C_{green},

2) a method for determining whether points are shadowed or unshadowed, and

3) "Correcting" the shadowed pixels using Equation (7).

Determining whether points were shadowed or unshadowed was accomplished by intensity threshholding. Approximate values for C_{red}, C_{blue}, and C_{green} were hand-selected by experimentation on several images containing

nant computes the line in RGB space such that when the pixel values are projected onto that line, the ratio of average interclass distance to average intraclass scatter is maximized. Intuitively, this results in the linear function which most cleanly separates the cut and uncut pixel classes. The discriminant function used for the first image is chosen arbitrarily; in practice, a poor choice may result in inaccurate crop line estimates for the first few images until the algorithm converges to more effective discriminant functions. Since current cycle times for our implementation of this algorithm are roughly 5 Hz, we have found the crop line estimates to be quite reliable 0.5 seconds after the crop line tracker begins to cycle.

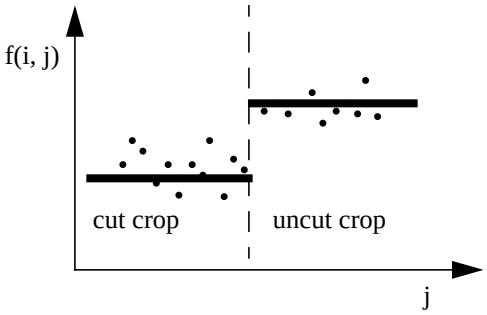

Figure 3: A model plot of f(i, j) as a function of j for a single scan line i.

A summary of our adaptive algorithm is given below:

1. Initialize the color discriminant function: f = 1.0 R+ 1.0 G + 1.0 B.

2. Digitize an image.

3. For each scan line in the image:

a. plot f as a function of image column i;

b. Compute the best fit step function to the above plot;

c. Return the location of the step as the crop line boundary estimate.

4. Compute an updated discriminant function using the Fisher linear discriminant.

5. go to step 2.

This algorithm allows for a very general crop line boundary; any single-valued function of image row can be represented. Figure 4 shows a typical result from this algorithm; in this image, a white dot has been placed on each scan line at the location of the estimated crop boundary.

Using the camera calibration parameters, each image row crop line boundary pixel is converted into a vote for a discretized pure pursuit steering angle. After the votes are tallied, the steering command with the most votes is relayed from the crop line tracker to the vehicle controller. In order to reduce processing time, low-resolution (160 x 120) images are used, and only that portion of the image

Figure 4: Sample output from the crop line tracker.

corresponding to the rectangle in Figure 4 is processed. Under these conditions, cycle times of 5-6 Hz are typical.

The use of the adaptive discriminant function improved performance considerably. Over time scales of a few minutes, or distances of a hundred meters, changes in the crop and soil color can be quite pronounced; however, the adaptive color metric deals fairly well with such variations. Highly localized differences in crop and soil color can be problematic; however, such variations within any single image are typically slight and relatively unstructured. Shadow noise presents a more serious challenge, and is discussed in detail in Section 3.

Using the adaptive algorithm, we have successfully tracked and cut entire curved crop rows of over 1 mile in length from circular fields in Kansas; this compares to maximum distances of 150 yards for the old non-adaptive algorithm using a discriminant of R/G. From one fairly challenging sequence of 29 images (from which Figure 4 is taken), the adaptive algorithm was able to correctly locate the crop boundary in 28 of the 29 images; by comparison, the R/G discriminant described in [8] was able to correctly locate the boundary in only 21 of 29 images.

3. Shadow Compensation

Shadow noise can heavily distort both image intensity (luminance) and color (chrominance). An example of a severe case is shown in Figure 5; here, a shadow cast by the harvester body lies directly over the region containing the crop line. This shadow is directly responsible for the resultant error in the crop line boundary estimate produced by the crop line tracker.

Shadow noise causes difficulties for a number of reasons. It is often quite structured, and thus is not well modeled by stochastic techniques. Its effects and severity are difficult to predict; if the sun is momentarily obscured by a passing cloud or the orientation of the harvester changes

Figure 1: The Demeter automated harvester.

control machine functions; a separate Sparc 20 is dedicated to the perception system. An on board GPS receiver coupled with a fixed base station allows the use of differential GPS-based positioning. Forward-facing RGB cameras equipped with auto-iris lenses are mounted to either side of the cab roof, near the ends of the harvester's cutter bar. The cameras are calibrated using a method developed by Tsai [11], which allows conversion of image pixel coordinates into real world locations.

One might reasonably question whether vision based guidance is necessary for this application. Automation of field vehicles using GPS (Global Positioning Satellite) appears promising; differential GPS systems with accuracies of 20 cm or better are commercially available at the time of this writing. While still expensive, these systems are rapidly falling in price as demand grows in applications such as surveying and automotive guidance.

As with all sensors, however, differential GPS is subject to various failures, such as satellite dropouts or broken communication links between the mobile and fixed GPS units. A vision system can provide an independent source of steering guidance while cutting; provide estimates to the end of the crop row; and detect potential obstacles. Further, such a system is extremely inexpensive compared to the cost of differential GPS. Rather than viewing vision based perception and GPS as competing sensor modalities, it makes more sense to consider GPS and vision as complementary; a combination of the two is likely to outperform either one alone.

In order to test the effectiveness of vision based sensing, we have performed a number of experiments using vision and dead reckoning as guidance for the harvester, without making use of the GPS system. A typical experiment begins by using the crop line tracker vision to follow the boundary between cut and uncut crop. The end of row detector is used as a trigger to decide when to transition into a turn behavior controlled by dead reckoning. When the turn is complete, the system transitions back to the crop line tracking behavior (Figure 2).

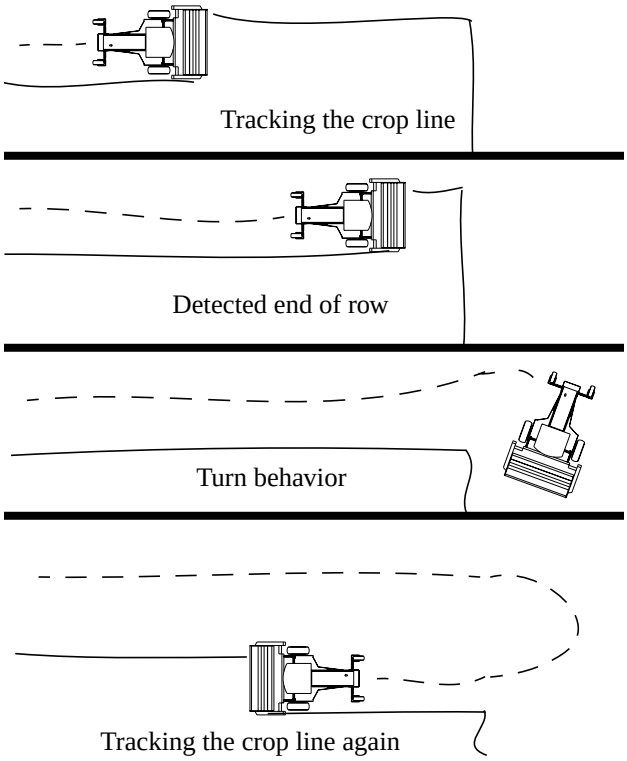

Figure 2: Transitioning between behaviors

2. Crop Line Tracking

The crop line tracking method used is an adaptive version of the algorithm presented by Ollis and Stentz [8]. Each scan line in the image is processed separately, in an attempt to find a boundary which divides the two roughly homogenous regions corresponding to cut and uncut crop. This is accomplished by computing the best fit step function to a plot of a pixel discriminant function f(i, j) for the scan line; the location of the step is then used as the boundary estimate for that scan line, as shown in Figure 3.

Previously published versions of this algorithm used a fixed discriminant function, such as f = G/(R+G+B). However, even within the same field, changes in lighting conditions and soil type prevent any single discriminant function from consistently returning a correct segmentation. To address this variability in the environment, we have implemented a method for adaptively updating the discriminant function.

After each image is processed, the algorithm computes the Fisher linear discriminant [3] in RGB space between the cut and uncut pixel classes; this becomes the discriminant used for the next image. The Fisher discrimi-

Vision-Based Perception for an Automated Harvester

Mark Ollis & Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213

Abstract

This paper describes a vision-based perception system which has been used to guide an automated harvester cutting fields of alfalfa hay. The system tracks the boundary between cut and uncut crop; indicates when the end of a crop row has been reached; and identifies obstacles in the harvester's path. The system adapts to local variations in lighting and crop conditions, and explicitly models and removes noise due to shadow.

In field tests, the machine has successfully operated in four different locations, at sites in Pennsylvania, Kansas, and California. Using the vision system as the sole means of guidance, over 60 acres have been cut at speeds of up to 4.5 mph (typical human operating speeds range from 3-6 mph). Future work largely centers around combining vision and GPS based navigation techniques to produce a commercially viable product for use either as a navigation aid or for a completely autonomous system.

1. Introduction

Agricultural applications have several appealing traits as candidates for automation. Current agricultural machinery is often expensive, so that sensing and computing can be added for a small marginal cost factor. The potential market size is large. Many agricultural tasks are dull, repetitive, and occasionally dangerous. They often take place in environments for which a priori knowledge is plentiful; for example, most agricultural machines only need to process one type of crop at a time, and accomplish their task within a known bounded geographic area.

This paper describes a vision-based perception system which has been used to guide an automated harvester through fields of alfalfa hay. Several vision based behaviors have been implemented as part of the Demeter automation project. A crop line tracker detects and follows the boundary between cut and uncut crop; an end-of-row detector estimates the distance to the end of the crop row; and an obstacle detector visually locates obstacles in the vehicle's path.

The seeming similarity of the road following problem to crop line tracking led us initially to attack this problem using RALPH [9], a highly successful road-following system. Results from an early RALPH experiment soon demonstrated several difficulties: for instance, ragged edges, highly variable curvatures, and uneven coloring were much more prevalent in the agricultural domain than in road following. We therefore investigated techniques developed explicitly for agricultural use.

Klassen and Wilson [6] describe an algorithm for distinguishing cut and uncut crop using a monochrome CCD camera. While their work was valuable in establishing that machine vision techniques could be applied to this task, there were some significant limitations: their system computes only straight line boundaries, requires a specialized digital signal processor, and has not been used to guide an actual vehicle. Gerrish et al. [4] used a variety of edge-detection and template-matching techniques to pick out "work edges"; these were tested on actual field images (including alfalfa). Straight line boundaries were still assumed, however, and processing times required were on the order of 20 seconds using a 68000 processor. Reid & Searcy [10] and Brandon & Searcy [2] have published work on a related problem, vision-based segmentation of crop canopy from soil, which further supported the applicability of vision-based techniques to this area.

Some results on guiding an actual agricultural vehicle are presented in Billingsley and Schoenfish [1], but their emphasis was on the discrimination of crop vs. soil for row crops, and they had limited opportunities for field trials. Ollis and Stentz [8] present a precursor to our crop line follower; subsequent work presented in this paper includes both a better algorithm for crop line following (including an adaptive capability and shadow compensation) and development of additional behaviors (end-of-row detection and obstacle detection).

The harvester is shown in Figure 1; it is a New Holland 2550 Speedrower retrofitted with wheel encoders and servos to control a number of machine functions, such as the throttle, steering and cutter bar. A Sun Sparc 20 board running a real time operating system (VxWorks) is used to