Engineering in Agriculture, Environment and Food 12 (2019) 24–31

Contents lists available at ScienceDirect

Engineering in Agriculture, Environment and Food

journal homepage: www.elsevier.com/locate/eaef

An overview of autonomous crop row navigation strategies for unmanned ground vehicles

Stephanie Bonadies[a,1], S. Andrew Gadsden[b,*]

[a] University of Maryland, Baltimore County, Baltimore, MD, 21250, USA
[b] College of Engineering and Physical Sciences, University of Guelph, Guelph, ON, N1G 2W1, Canada

ABSTRACT

Unmanned ground vehicles (UGVs) are becoming popular for use in agricultural environments. These unmanned systems are implemented in order to address human labor shortages throughout the agricultural industry, and improve food safety throughout the production cycle of produce crop. Common uses of UGVs in agriculture include: detection of animal fecal matter, surveys of crop growth, detection of crop damage from storms or floods, and detection of unwanted pests or molds. Navigation of crop rows is typically accomplished using vision-based cameras and global positioning system (GPS) units. Machine vision strategies are implemented to detect crop row contours and edges to ensure proper navigation of rows without damaging crops. A number of other control and navigation strategies exist for autonomous movements of UGVs. This paper provides a survey and overview of autonomous navigation strategies for UGVs with applications to agricultural environments.

1. Introduction

The development of UGVs for various uses from agriculture operations to military operations has been occurring for the last few decades (Sistler, 1987). Improving the efficiency of agricultural production is a concern as the world population continues to increase. The unique ability of UGVs to travel through fields, while supporting sizable payloads makes them ideal for an agricultural environment. The development of UGVs for particular agricultural applications is ongoing in academia and the engineering industry to combat issues of labor shortages and foodborne illness which affect the agricultural industry today (Hamrita et al., 2000). Many of the currently available commercial UGVs do not offer autonomous navigation for agricultural environments, and instead depend on remote control (Husky Unmanned Ground Vehicle Robot, n.d.; Jackal Small Unmanned Ground Vehicle, n.d.). Various researchers have developed autonomous UGVs for specific agricultural tasks in crop rows, however these systems are designed only for that particular application (Lefcourt et al., 2016). Commercially available or open-source hardware and software are not available for crop navigation.

Currently, GPS and geographic information system (GIS) are the most commonly used means of guiding vehicles through an agricultural environment without the input of a human operator. In general, these vehicles rely on pre-planned routes, rather than having the ability to operate in new or changing environments. In order to ensure high location accuracy, precision GPS receivers are used. Real-time kinematic (RKT) and Differential GPS (DGPS), which use reference stations located in the target environment, are used to enhance the accuracy of GPS signal down to a few centimeters. The autonomous weeding robot developed by Bakker et al. (2010a) navigates using this type of technology, relying on two GPS antennas connected to a RTK-DGPS receiver, improving location accuracy to 1–2 cm (Bakker et al., 2010a).

An alternative to GPS navigation is to use sensors or cameras to interpret the local environment as the UGV travels. Radar and ultrasonic sensors can be used to detect large obstacles or landmarks, and can be used in well controlled indoor environments or roads with predefined surroundings for navigation. However, in uncontrolled outdoor environments, natural variability can test the limits of these types of sensors. Light Detection and Ranging (LIDAR) systems can also be used to produce either a 2D or 3D rendering of the surroundings. LIDAR systems utilize a laser or combination of lasers, and are not limited by visibility or ambient light levels.

While 3D LIDAR sensing systems can be expensive, cheaper 2D sensors can be manipulated to produce a 3D rendering for use in detecting static and dynamic objects in the surroundings of a UGV, as discussed by Rejas et al. (2015). A 2D laser sensor mounted atop a continuously rotating platform was used to emit a focused laser beam and receive and interpret reflection levels of objects in the path of the laser. Based on this information, the distance between the robot and an obstacle or object of interest was determined. The rate of rotation of the

* Corresponding author.
E-mail address: gadsden@uoguelph.ca (S.A. Gadsden).
1 Current affiliation: Northrop Grumman Corp., Baltimore, MD, 21090, USA.

https://doi.org/10.1016/j.eaef.2018.09.001
Received 27 July 2017; Received in revised form 1 April 2018; Accepted 2 September 2018
Available online 10 September 2018

laser was adjusted by the system based on the density of the objects detected in the path of the laser and the vehicle speed. Using a Hokuyo 30LX LIDAR system with a detection angle of 270° and a detection range of 0.1–30 m, the system was used to create a 3D image of a room containing the scanner. The software for this system was developed in ROS for optimal processing speed. Using a rotation speed of 1.5 RPS, the scanner can be mounted atop a vehicle travelling up to 5.5 m/s to avoid obstacles in its path. After various experimental trials, it was determined that the rotation speed of the scanner should be adjusted based on the speed of the vehicle for optimal results. The system was able to detect obstacles between 0.1 and 10 m surrounding the UGV, and could be used to produce 3D map renderings of the vehicle's environment as it travels.

Camera vision can also be used to visually interpret the environment for navigation and obstacle avoidance. Cameras can be used to produce 2D images or 3D renderings of the local environment. A standard camera can be used to capture 2D still or video images of the crop environment, which can be processed for guidance. Navigation based on image processing utilizes images as an input signal to detect edges of crops and rows by recognizing differences in color or shape, or avoid obstacles while travelling through a field. Imaging relies on a light source to induce reflectance or a fluorescence response of the area of interest. In agricultural fields, sunlight or infrared light sources can be used to induce reflectance for data acquisition, while various light sources including ultraviolet light can induce a fluorescent response. Image processing can be used to navigate an unknown environment and respond to changes in real-time (Mousazadeh, 2013).

In contrast, stereo vision cameras are capable of creating a 3D image of the environment by imitating human vision, which combines data from two separate images of the same scene to interpret the depth of various objects. Navigation based on stereo vision through agricultural crop rows requires a detectable height difference of the crop above the ground (English et al., 2014). In earlier growing stages, crops may too short to provide enough information for navigation based on the height of the crop. Additional image processing is needed to distinguish between the crop and ground compared to mono vision cameras. Since the processing of the stereo images required to determine depth information is significant, this method may be less efficient than the use of mono vision cameras to collect 2D images of the environment (Rejas et al., 2015).

English et al. (2014) developed a guidance method for an autonomous weed spraying platform utilizing two different cameras for image capture: an IDS uEye CP and a low-cost Microsoft LifeCam Cinema webcam. A row-tracking control system developed in C++ using OpenCV was implemented on the platform. The image processing algorithm performed a series of processing and calculation steps on the image collected by the camera. First pre-processing was performed to remove lens-distortion effects, then the location of the horizon was detected to determine roll and pitch. Next, the image was straightened using roll and pitch estimations, and the image was warped to create an overhead view for interpretation. Using the warped image, an estimate for vehicle location and heading were determined. After correcting for robot heading angle from the desired track was determined. Using a Proportional-Integral controller, closed-loop experiments were run to observe the vehicle's ability to autonomous navigate wheat and sorghum stubble rows. For comparison, RTK-GPS data was collected to record vehicle position throughout the experiments. The RMSE between the vehicle position and the true lane location for these trials was 28 mm and 120 mm, respectively. Despite noisy images in some locations, the vehicle was able to successfully navigate both crop rows without modification to the image processing algorithm.

Similarly, Xue and Xu developed a vision-based row guidance method for an autonomous weeding and spraying robot (Xue & Xu, Autonomous Agircultural Robot and Its Row Guidance, 2010). This vehicle was equipped with a Sony CCD camera mounted to the front of the platform. The row detection method relied on the detection of two crops in front of the robot. Calculating the center of the detected plants, and drawing a line between these points, the position and heading of the vehicle was determined by comparing to a predefined line at the center of the image. The image processing algorithm used edge detection and predetermined information about the crop size and distance between each crop to detect plants in the image. The collected image was converted to a binary image, and after detecting the contours of the plants in the image, the plant's centroid was determined. These two points were used to create the guidance line for navigation. Using a fuzzy logic controller with two inputs (vehicle position and heading) and one output (motor output commands), the vehicle was run through a small vegetable crop lane. The true position of the vehicle throughout the trial was recorded using a painted line drawn by the robot as it travelled. With the vehicle travelling at 0.2 m/s, it was able to maintain a position within ± 35 mm of the desired path.

Takagaki et al. (2013) created an image processing method that could be used for autonomous navigation of a ground vehicle through agricultural environments based on ridge and furrow detection between rows. Two different image processing algorithms were used for images with shade and images without shade, as determined by analysis of gray level histograms. In order to navigate through rows with shade, color differences were used, while texture detection was used to navigate when shade was not present. For shade images, shadows were present on one side of the ridge based on the angle of the sun. The edge of a ridge was determined by observing the pixel value difference between the light side of the ridge and the shaded side of the ridge. After the edges of the ridges to the left and right of the vehicle were determined, a Hough transform was applied to determine the equations of these lines, and a center line between the two was calculated for vehicle navigation. For images without shade, the variance within a square region was observed to find the area in the image where the soil is smooth, indicative of a furrow. Using the minimum variance values, the furrow can be distinguished from the ridges in a binary transformation. Finally, a Hough transform is applied to obtain the equation of the furrow line for guidance along the row.

Testing the algorithms in four different fields, the camera was used to acquire images with a variety of lighting conditions. The image sorting algorithm correctly distinguished the shaded images from those without shade 100% of the time. Of the 30 images taken with shade, the image processing algorithm successfully determined the row center 100% of the time. Of the 23 images without shade, the image processing algorithm successfully determined the lane center 87% of the time. The image processing algorithm, Takagaki et al. predict, could be implemented into a control system to guide an UGV through the rows (Takagaki et al., 2013).

Navigation in an agricultural field may require not only navigating through a crop row, but also recognizing the start and end of a row, and turning between rows. While image or sensor data can interpret when an UGV has reached the end of a row, additional control is needed to instruct the vehicle to make turns and locate the next row. Based on the size of the rows in a particular field as well as the size of the UGV, software can be programmed to enable the robot to make a predetermined turning maneuver when the end of a row is reached.

The purpose of this paper is to provide an overview of autonomous navigation strategies for UGVs with applications to agricultural environments. The paper is organized as follows. Navigation techniques for UGVs are described in Section 2. Two of the most popular lane detection algorithms are summarized in Section 3, and include crop row detection using contour tracing and Hough line transformation. Section 4 describes lane detection based on popular control strategies: PID and fuzzy logic. The paper is then concluded in Section 5.

2. Navigation techniques

In order to control the motion of an UGV, manual or autonomous control can be used. A manual control scheme can be utilized to direct a robot in performing desired tasks. For mobile, UGVs, manual control can be executed by synching a wireless handheld controller, mobile device or tablet to the vehicle motion. Control actions may be sent to the vehicle from close proximity or from a remote location, depending on the functionality of the wireless communication. Cameras may be mounted atop the vehicle to stream live feed images to the operator (Yang et al., 2015). Instead of using manual control, various automated control schemes can be implemented to enable autonomous navigation in an UGV. In order to accomplish autonomous motion, the position or location (globally or relative to its surroundings) of the vehicle must be determined and interpreted as an input. Next, the input can be processed to determine a navigational goal (e.g. moving forward, backward, turning, or stopping). Finally, this goal can be outputted as information to the vehicle motors and wheels. Methods of positioning and localization include dead reckoning, range sensing, reflectance sensing, and image processing (Xue, Zhang, & Grift, Variable Field-of-View Machine Vision Based Row Guidance of an Agricultural Robot, 2012).

Dead reckoning is a positioning method which calculates vehicle position based on the distance, angle, and speed of travel. While the positioning is accurate at first, as the travel time increases, accumulation of error from slipping decreases accuracy (Mousazadeh, 2013). GPS information can be used for autonomous vehicle navigation by determining absolute vehicle location; however, the localization information is limited by the GPS receiver. Low-cost GPS sensors can provide accuracy to within meters, while more expensive receivers can provide accuracy to within centimeters (Patel, 2015). Range sensing can be accomplished by implementing infrared, ultrasonic, or other sensors on a vehicle to detect the distance from an object for obstacle or landmark detection (Mousazadeh, 2013). In a similar fashion, reflectance information from a photo resistor can provide information about changes in the surrounds. The photo resistor will pick up varied inputs based on the amount of light reflected off of a particular object. Reflectance of plants will be different than that of dirt or other objects in the field (Bakker et al., 2010a). Navigation based on image processing utilizes images as an input signal to detect crop rows or avoid obstacles while travelling through a field. Image processing can be used to navigate an unknown environment and respond to changes in real-time (Mousazadeh, 2013).

While the information gathered from these sensors and methods alone may be enough to determine an appropriate output to the vehicle, control algorithms can be implemented to improve the accuracy and efficiency of vehicle movement. For example, to assist in converting input information into viable navigational control, various algorithms can be implemented including: proportional-integral-derivative (PID) control, fuzzy logic control, neural networks/genetic algorithm control, and Kalman filtering. Mousazadeh (2013) conducted a review of control algorithms used in various agricultural ground vehicle applications, concluding that a combination of multiple control strategies would provide the best approach to navigation.

While a variety of UGV platforms are commercially available for research and development, software for autonomous motion in crop rows is not. Many UGV platforms are equipped with GPS antennas for navigation of environments based on predetermined paths. However, in a row crop environment, navigation via GPS may not provide the accuracy needed to avoid damage to crops, or may require the addition of antennas to the field for centimeter-accuracy, depending on the location. In these cases, it may be beneficial to explore the use of image-based navigation for real time interpretation of the UGV's current environment.

3. Lane detection algorithms

Control strategies have been developed to allow robots to navigate crop rows autonomously using machine vision (Xue, Zhang, & Grift, Variable Field-of-View Machine Vision Based Row Guidance of an Agricultural Robot, 2012). The image from the camera will be interpreted by a guidance algorithm to distinguish between the crops and the soil between the crops. Once the location of the edges of the crops on the left and right sides of the robot are determined, a center point, representing the location of the center of the row within the image, will be determined, as illustrated in Fig. 1. A predetermined setpoint representing the value of the center point when the robot is centered within the lane will be defined based on the image collected by the selected UGV. The error between the calculated center point and the predetermined setpoint will be fed through either the PID (Method #1) or the Fuzzy Logic (Method #2) controller to determine the movement necessary to move the robot to the center, reducing the error to zero.

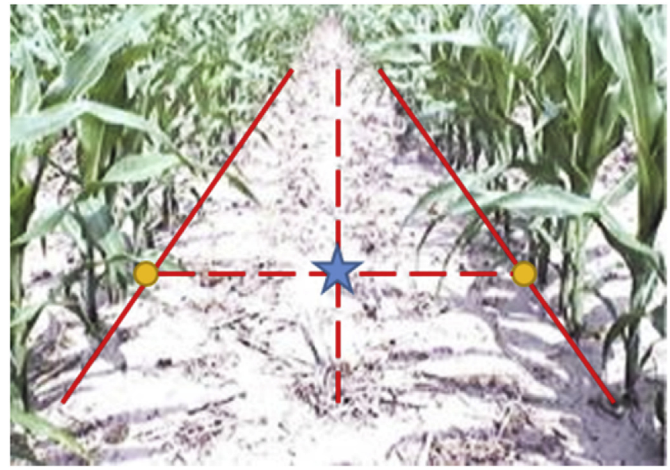

Fig. 1. Row center determination illustration (Xue, Zhang, & Grift, Variable Field-of-View Machine Vision Based Row Guidance of an Agricultural Robot, 2012).

The crop lane detection algorithm requires processing of the camera image to determine which parts of the image are crops as opposed to rows. Two image processing approaches are explored for detecting the edges of crop rows: one using contour detection, and another using line detection. Common techniques used to detect a particular color in images include binary segmentation, applying morphological filters for noise reduction, obtaining an image complement to flip the black and white pixels in the binary image, and tracing contours within the image. Alternatively, a Hough transform (Hough Transformation, 2016) can be used to detect the lines after an edge detection filter has been applied to the image. If the rows of crops can be detected accurately and efficiently, navigation through the row based on image data may be possible.

3.1. Crop row detection using contour tracing

Since a stark contrast exists between the soil and the crop rows for food crops, particularly leafy greens and other produce which make up a large portion of the consumables that lead to foodborne illness, common image processing techniques can be used to detect the edges of the rows in real-time. For example, a lettuce field is dominated by large rows of green plants alternating with brown dirt lanes (Fig. 2). Processing this image can lead to the detection of the lanes between the rows for effective navigation.

In order to begin processing the image, it is converted to grayscale as a preprocessing step (Demant et al., 2013). The 8-bit image is made up of three masks, one for red, green, and blue (RGB), respectively. The colors are determined by a combination of three brightness values, each corresponding to one of these RGB masks, with values ranging from 0 to 255. If one of the three masks is isolated, the original brightness from 0 to 255 corresponding to a particular pixel for the color selected will be used to convert all three RGB values to the same level corresponding to

Fig. 2. Sample lettuce crop row (Rows of Lettuce, n.d.).

this brightness. When all three RGB values are equal, the corresponding color falls between white (maximum brightness) and black (minimum brightness), storing only the intensity information, thus the pixel will be in grayscale.

For example, in this case, the green mask is of interest since the region of interest (lettuce crop) is green, while other areas of the field are brown. Isolating the green mask of the image, which is the second layer of the RGB file, will form a grayscale image corresponding to the intensity of green in the image. As can be seen in Fig. 3, the top image displays the original crop rows with the RGB information for a point within the crop row at (300, 210). The grayscale RGB values are represented as a percentage between 0 and 1, calculated by computing the quotient of 171, the green value, over 256, which is equal to 0.667. The greater the green index is at a particular point, the greater the value of the quotient, and thus the brighter the grayscale pixel will be. Pixels with a greater intensity, closer to 255 (or 100%), are more green, and appear to be more white in grayscale. Pixels with a lower intensity, closer to 0, are less green, and appear darker in grayscale.

Since the green objects in the image will appear as brighter pixels, a brightness threshold can be set to distinguish between the areas in the grayscale image, converting the image to binary. In this case, a global threshold, which uses a single gray level for the entire image, can be set (Demant et al., 2013). A global threshold will define all gray values above the threshold value as white, and all those below the threshold as

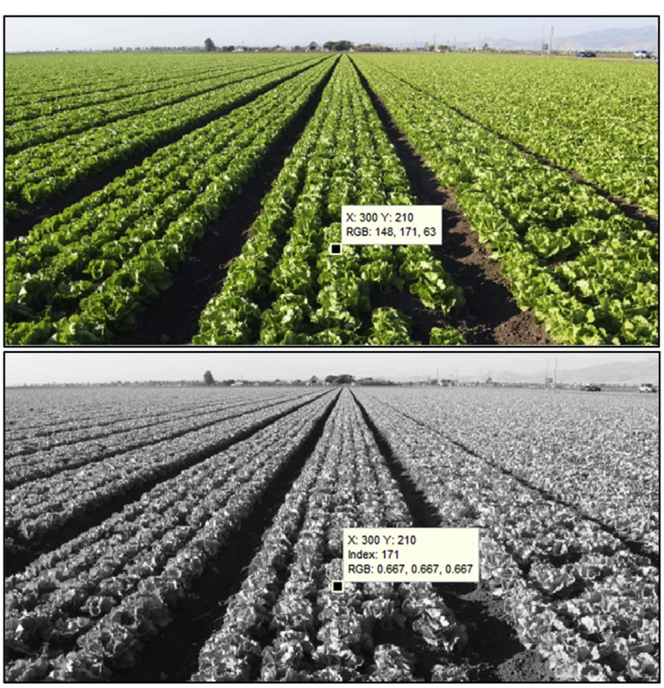

Fig. 3. Isolating the green mask of the lettuce rows.

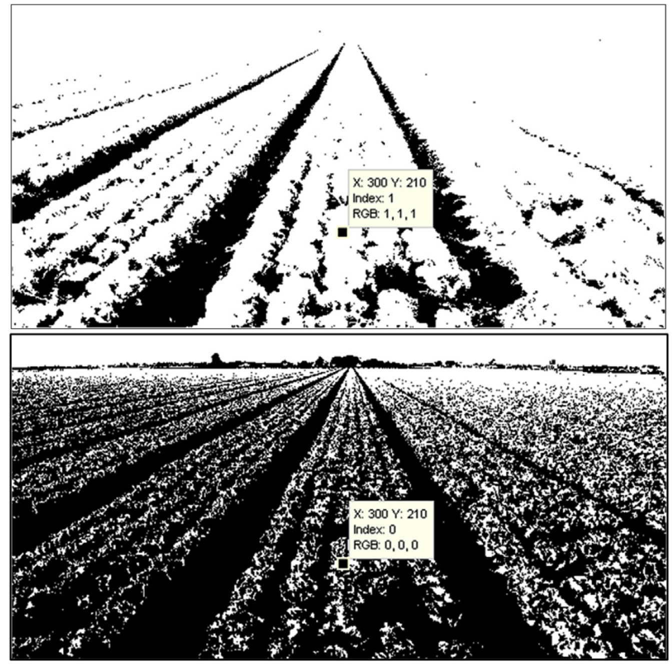

Fig. 4. Binary Images based on Threshold Value of 50 (top) and 172 (bottom).

black. For example, if the threshold value is set to 50, the point (300, 210) will be converted to a white pixel, with RGB value (255, 255, 255). However, if the threshold is set to 172, just above the original green value of 171 from the original image, this same pixel is converted to black, with RGB value (0, 0, 0), since it falls below the set threshold (Fig. 4). In this case, a value of 50 is a more appropriate threshold for this image.

Now that the image is binary, the contours of the lanes between the crop rows can be traced. The contour of the lane is the connected line which encloses all of the pixels that make up the object. In order for the program to detect a contour, the concept of connectedness must be defined. Two basic types of connectedness are used for 2-D images: four-connectedness and eight-connectedness (Fig. 5). Starting with a single point, the pixels surrounding it can be considered connected only if they fall next to this point in the horizontal or vertical directions when using four-connectedness. In contrast, diagonal pixels are also considered to be connected to the point when using eight connectedness.

Based on the selected connectedness definition, a contour detecting algorithm can be developed using the following steps for lane detection. First, a search is initiated for a transition between the rows and the lanes between the rows. Once this transition is located, the next neighbor, based on the connectedness definition is determined. Since the contour represents the border or perimeter of the object, only pixels

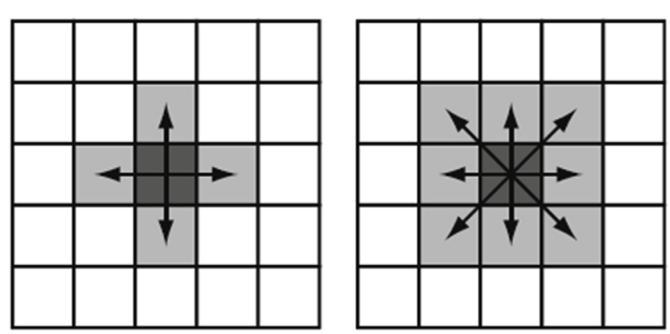

Fig. 5. 4-connectedness (Left) vs 8-connectedness (Right) (Demant et al., 2013).

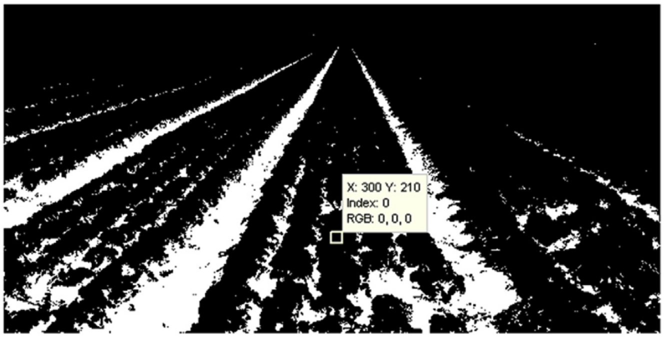

Fig. 6. Inverted Binary Image based on Threshold of 50.

Fig. 8. Grayscale image from the green mask of the example image.

along the object border are considered neighbors. Moving in a clockwise or counterclockwise direction, the entire image is searched until the contour is traced. The contour has been completely traced when the algorithm reaches the point at which it started, signifying a closed region.

In general, the object of interest in a binary image is considered to be that which is represented by white pixels (pixels with a value of 1), while black pixels (value of 0) are considered the background. In this example, the crop rows are the area of interest that is denoted by white pixels, but the lanes between the rows are the contours that should be detected for row guidance. As such, the image can be inverted, so that all of the pixels representing the crop rows are flipped from a value of 1 to a value of 0 and all of the pixels representing the lanes are flipped from 0 to 1. Considering the threshold value of 50, the image is inverted, and the point (300, 210) is now black, while the lane pixels are white (Fig. 6).

In various software platforms, such as MATLAB, predefined functions are available to detect contours without having to program the algorithm. In MATLAB, the function "bwconncomp.m" can be used to find all of the connected objects within a binary image. This function stores the x-y coordinates of all of the points that make up the contour, which can later be used to calculate the center of a lane or row. After the contours are detected in the image, they can be displayed on the image. In the row detection case, the lanes between the rows should be the largest contours in the area of interest (Fig. 7). The two largest contours in this image are the lanes to the left and right of the center crop row. Based on the shape of the contours, the left lane is better detected than the right lane because of the shadows within the field. The lane detection can be improved in a variety of ways including filtering the binary image to remove noise and tuning the threshold value to adapt to the ambient light in the field.

3.2. *Crop row detection using hough line transformation*

In contrast to row detection, the lines which follow the edges of the crop row can be determined using Hough transformation (Hough Transformation, 2016). Similar to the previous method, the first step is to isolate the green mask of the image (Fig. 8). Next, an edge detection algorithm can be used to convert the image to binary based on the shapes in the image. For example, Canny edge detection, which combines five steps, smoothing, finding gradients, non-maximum suppression, double thresholding, and edge tracking by hysteresis, can be used to distinguish between edges and the rest of the image (Canny Edge Detection, n.d.). The output image from the Canny edge detection algorithm is a binary image, with the edges of objects as white pixels, and the rest of the image as black pixels (Fig. 9).

Fig. 7. Contours in lettuce row.

Once the edges of the objects within the image have been detected, a Hough transform can be performed to determine the equation of the lines within the image. In MATLAB, the "hough.m" function can be used to implement a Standard Hough Transform, using the parametric representation of a line (Equation (1)):

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{1}$$

where ρ is the perpendicular distance from the image origin to the line of interest in the image and θ is the angle, ranging between −90 and 90°, between the perpendicular projection and the x axis (Fig. 10).

The inputs to the "Hough" function are a binary image, and optional rho and theta resolution values. The "Hough" function outputs the Standard Hough Transform (SHT), which is a parameter space matrix with rows and columns corresponding to rho and theta respectively, in addition to the rho and theta arrays. The values in the SHT are the number of points that lie on a line that is specified by the particular rho and theta. The peak values in this matrix signify potential straight lines within the image.

The peak values in the SHT matrix can be determined using the "houghpeaks.m" function, which uses the SHT matrix and a specified number of peaks to look for. The output of "houghpeaks.m" is a matrix containing the rho and theta values of the specified number of peaks. Finally, the rho and theta values of the peaks can be used to calculate the equations of the lines in the image using "houghlines.m" based on the formula in Equation (1). These lines can be drawn in the image for visual comparison with the crop rows. For example, using the example

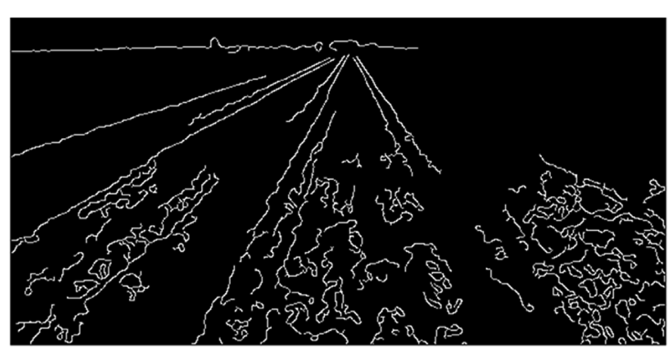

Fig. 9. Binary image output of canny edge detection algorithm.

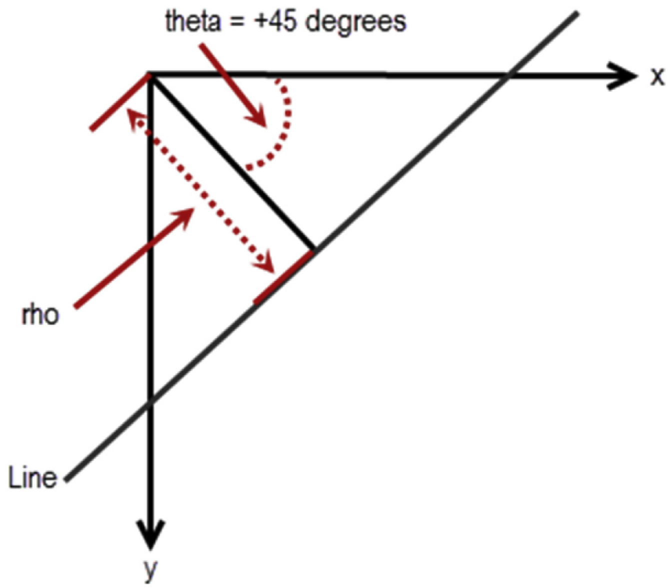

Fig. 10. Graphical representation of ρ ("rho") and θ ("theta") from the hough transform (Hough Transformation, 2016).

Fig. 11. Hough transform lines in the example image.

image, Hough transform was able to detect two straight lines along the edges of the center crop row (see Fig. 11).

It can be noted that while the line left side of the crop row follows the edge of the row well, the line along the right side seems to be angled slightly off from the actual row. Utilizing filters to remove noise from the image before performing the Canny edge detection, as well as adjusting the Hough transform configuration can be useful in improving the line detection for this particular image. In addition, the Hough transform did not detect both edges of the lanes between the rows, but rather only detected the edges of the largest row. A limitation of this particular method is that the resulting lines are straight line approximation of the crop row edges. Based on the curvature of the particular row, this straight line approximation may not convey the necessary information for successful navigation. Selecting a smaller region within the image collected in which to perform the Hough transform to determine the row edges would improve the approximation.

4. Lane detection with control strategies

Based on the two lane detection techniques described above, two approaches can be taken to guide a UGV through the crop rows. For one, the UGV can be configured to follow the edges of the lane between two crop rows. Another approach would be to configure the UGV to detect the edges of the crop row itself. Since the wheels (or tracks) of the vehicle will be on the ground between the crop rows, damage can occur if the wheels (or tracks) begin to roll over the crops themselves rather than over the soil between the rows. Using the image data to the detect the crop rows or lanes between the rows can be used in combination with a PID or Fuzzy Logic controller to ensure that the UGV stays within the desired lane to avoid damage to the crops.

The configuration of the vehicle platform will determine which approach is appropriate for this project. For example, if the vehicle is to use the edges of the crop row for navigation, it will need to take an image of the entire row. If the UGV is much shorter than the crops in the row, the camera image may be blocked by the surrounding crops. To overcome this, a platform can be used that is tall enough to mount the camera above the row. Alternately, a platform that can straddle the crop row, with one side of the vehicle in the lane on the left and the other in the lane of the right side of the crop row could be used. In contrast, an UGV which uses the edges of the lane between crop rows can navigate, regardless of whether it can see over the tops of plants. Both UGVs that fit within a lane and UGVs that straddle an entire crop row could utilize the lane detection approach. Since the lane detection approach seems to be better suited for a wider variety of UGV platforms, controllers using approach are outlined below.

To convert the lane detection data to commands for the motors for UGV navigation, an area of interest within the image can be determined. For example, based on the UGV speed, the transmission speed of the image to the computer that processes the image, and the transmission speed of the navigation commands to the motors, navigation data at a certain distance in front of the image is needed in order for movement correction to be applied in time. In a 2-D image, this area of interest can be selected as a set pixel value on the y-axis, which represents a particular distance from the front of the UGV.

In addition, the center of vehicle relative to the image at this area of interest should be known. This can be determined through a calibration even before developing the controller. Theoretically, each time the vehicle is centered and facing forward, the center of the row being detected should align to the same pixel value. Centering the vehicle within a sample row and processing a series of images taken by the robot at this location can be used to determine the pixel value of the row center. Depending on the accuracy of the image processing algorithm, this center pixel value should be applicable across rows of many widths, and can used as the set-point.

4.1. Method #1: lane detection with PID control

One of the oldest and most commonly used forms of system control is PID, which combines proportional, integral, and derivative terms. PID control requires a user to tune three constant parameters to control the system response: k_p, k_i, and k_d. Increasing the k_p term generally improves the system rise and settling times, while increasing k_i improves the steady state error, and increasing k_d improves the percent overshoot. However, changing each of the gain terms can also negatively impact the system response characteristics, as summarized in Table 1. For example, increasing k_p can increase the percent overshoot of the response. As such, fine tuning to determine the optimal combination of the three gain terms is important.

The error between a measured system variable and a desired set-point is continuously calculated and is used to calculate an output to a process variable. Equation (2) displays the general form of a PID controller:

Table 1
Impact of increasing PID gain terms on system response (Introduction: PID Controller Design, n.d.).

Gain Term	Rise Time	Overshoot	Settling Time	Steady State Error
k_p	Decrease	Increase	Small Change	Decrease
k_i	Decrease	Increase	Increase	Eliminate
k_d	Small Change	Decrease	Decrease	No Change

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\mathrm{d}\tau + k_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \quad (2)$$

where $u(t)$ is the controller output variable and $e(t)$ is the error between the desired location of the vehicle (set-point) and the actual location of the vehicle based on the image data acquired. The derivative and integral values can be approximated in discrete time for implementation in control software as seen in Equations (3) and (4):

$$\frac{\mathrm{d}e(t)}{\mathrm{d}t} \approx \frac{[e(t) - e(t-1)]}{T} \quad (3)$$

$$\int_0^t e(\tau)\mathrm{d}\tau \approx [e(t) - e(t-1)]\cdot T \quad (4)$$

where t is the current time step, $t-1$ is the previous time step, and T is the sampling rate. PID controllers are used for a variety of industrial applications temperature control in furnaces and pH regulators (Common Industrial Applications of PID Control, n.d.). PID control has also been used for agricultural applications for driver assistances. Foster et al. utilized a PID controller to autonomously regulate the velocity of a hydrostatic windrower to improve the machine productivity (Mousazadeh, 2013).

The implementation of a PID controller requires gain tuning in order to assist in guiding the system input to a predetermined set-point. The following steps outline the software algorithm for lane detection using a PID controller (Fig. 12):

In Step 1, a camera facing forward connected to the UGV will be instructed to capture an image of the area. This image will be stored within the software, and then processed in Step 2 by the image processing algorithm. The goal of the image processing algorithm will be to determine the pixel value of the crop row on both the left and right sides of the robot at the predetermined area of interest (y-axis pixel value). With the pixel values of the crops rows surrounding the UGV, Step 3 can be completed to calculate the pixel value of the center of the crop lane. Then, the set-point pixel value can be subtracted from the lane center pixel value to determine the error from the set-point in Step 4. Once the error is determined, it can be feed through a PID controller with predefined gains in Step 5 to determine appropriate motor output values. Then in Step 6 the PID output motor values can be sent to the motors. Finally, in Step 7, the algorithm is instructed to return to Step 1 and repeat the loop.

Setting up the PID controller for navigation will require manual gain tuning efforts for the real system. The goals of this tuning effort should be to reduce the percent overshoot of the system response to avoid running into crops after a correction. The desired maximum percent overshoot will vary based on the width of the crop lane that the vehicle travels down, as a higher overshoot may be allowable within wider rows. In addition, reducing the settling time of the system such that the distance between the center of the UGV and the lane center is within a predetermined error from center (± 1 or ± 2 inches) will maximize the amount of time the robot is travelling forward at maximum speed.

4.2. Method #2: lane detection with fuzzy logic control

A Fuzzy Logic controller utilizes fuzzy logic to produce desired outputs based on given inputs. Fuzzy logic, in contrast to Boolean logic, allows for varying degrees of truthfulness between 0 and 1, rather than absolute truth and falsity. In order to design a fuzzy controller, membership functions must be developed for the system input and output, coupled with a set of rules to handle the inputs and determine what output is appropriate for the current state of the system (Gerla, 2005).

A Fuzzy control system has three parts: fuzzification, rule evaluation, and defuzzification. A set of crisp inputs, for example sensor input data, is transformed into a set of fuzzy inputs through fuzzification. A set of input membership functions, which encompass the relationship between all possibly input values, is used to convert these sensor input values to a fuzzy input value ranging between 0 and 1. Developing appropriate membership functions for the input set is important; using too few can lead to slow system response and using too many can cause instability in the system. After the crisp inputs are converted to fuzzy inputs, these values are fed through a set of rules developed for the system. These rules are used to determine the controller output based on the sensor input data in the form of an IF-THEN statement, which relates the output (dependent) variables to the input (independent) variables. Based on the fuzzy input values, the rules are evaluated and the rule that is most true is used to determine the fuzzy outputs. Finally, the fuzzy outputs are converted into crisp outputs through defuzzification, which requires a second set of membership functions, converting the fuzzy outputs between 0 and 1 to meaningful output values.

Fuzzy Logic control has been used for steering control in agricultural and military robots. A Fuzzy Logic controller was implemented into the DORIS robot at the University of Germany to control the commands sent to the motors based on the steering wheel angle and the level of force applied to the break or gas pedal to ensure smooth turning maneuvers (Sailan et al., 2014). In addition, an UGV developed by Xue et al. utilized a Fuzzy Logic controller to guide a robot through a corn crop row based on the detected location of crop rows on either side of the robot. The Fuzzy controller used two inputs (offset from center line and heading angle), fuzzified by five triangular membership functions with a uniform distribution. The input information was compared to the previous position of the robot, and various turn signals were outputted to the motors based on the position and heading angle (Xue, Zhang, & Grift, Variable Field-of-View Machine Vision Based Row Guidance of an Agricultural Robot, 2012).

The implementation of a Fuzzy Logic controller requires calibration and development of membership functions for both the input and output variables of the system. The following steps outline the software algorithm for lane detection using a Fuzzy Logic controller (Fig. 13):

Similar to the PID controller, in Step 1, a camera facing forward connected to the UGV will be instructed to capture an image of the area. This image will be stored within the software, and then processed in Step 2 by the image processing algorithm. With the pixel values of the crops rows surround the UGV, Step 3 can be completed to calculate the pixel value of the center of the crop lane. Then, the set-point pixel value can be subtracted from the lane center pixel value to determine the error from the set-point in Step 4. This step can be eliminated if the membership functions are calibrated to directly interpret the lane center information. Once the error is determined, it can be fuzzified using predetermined input membership functions for the system in Step 5. In Step 6, the fuzzified error value will be evaluated using the rule set for the Fuzzy Logic Controller, and used to determine fuzzified output motor values using the output membership functions. Then in Step 7 the Fuzzy output motor values can be defuzzified to be sent to the motors. The defuzzified motor command values can be sent to the

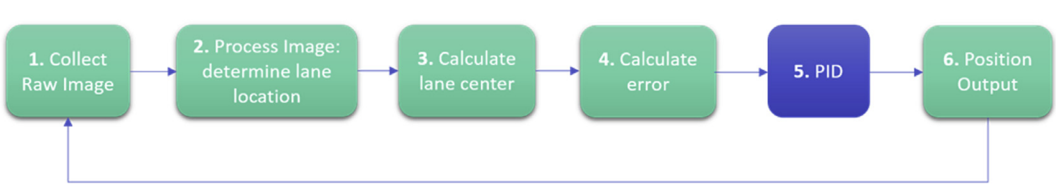

Fig. 12. PID controller lane detection.

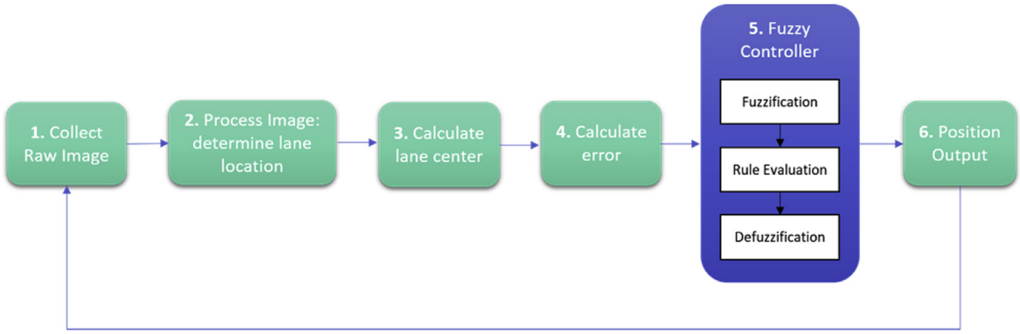

Fig. 13. Fuzzy controller lane detection.

motors in Step 8. Finally in Step 9, the algorithm is instructed to return to Step 1 and repeat.

In order to utilize the Fuzzy Logic controller, a calibration step must be performed to determine the values of the membership functions for the input and output variables. Calibration for the image input variable can be completed by setting up the UGV at different, known locations across the front of the crop lane to determine what range of error values corresponds to the physical position of the robot within the lane. The rules to evaluate the input should be setup such that the UGV performs certain turning maneuvers based on the input from the image data. Predetermined set speeds can be used to set up the membership functions for the output variables to enable the UGV to perform the various turning maneuvers to return to the center of the crop lane. Finally, the fuzzy output should be converted to a command value which is meaningful to the motors so that the maneuver can be performed.

5. Concluding remarks

Unmanned systems such as UGVs are implemented in order to address human labor shortages throughout the agricultural industry, and improve food safety throughout the production cycle of produce crop. The most common use of UGVs in agriculture is detecting contaminated plants and crops, searching for the presence of animals or pests, and identifying crop funguses and molds. UGVs navigate crop rows, and as such, autonomous navigation strategies have been developed. These strategies typically make use of machine vision and PID and fuzzy control methods. The purpose of this paper was to provide a comprehensive overview of autonomous navigation strategies for unmanned ground vehicles (UGVs) with applications to agricultural environments.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.eaef.2018.09.001.

References

Bakker, T., Asselt van, K., Bontsema, J., Müller, J., Straten van, G., 2010a. Systematic design of an autonomous platform for robotic weeding. J. Terramechanics 47 (2), 63–73.
Bakker, T., van Asselt, K., Bontsema, J., Müller, J., van Straten, G., 2010b. A path following algorithm for mobile robots. Aut. Robots 29 (1), 85–97.
Canny Edge Detection, (n.d.). (OpenCV). Retrieved from: http://docs.opencv.org/master/da/d22/tutorial_py_canny.html#gsc.tab=0.
Common Industrial Applications of PID Control, (n.d.). (Control Station Inc.). Retrieved from: http://controlstation.com/pid-control/.
Demant, C., Garnica, C., Streicher-Abel, B., 2013. Overview: segmentation. In: Industrial Image Processing: Visual Quality Control in Manufacturing. Springer, Berlin, pp. 83–112.
English, A., Ross, P., Ball, D., Corke, P., 2014. Vision based guidance for robot navigation in agriculture. In: 2014 IEEE International Conference on Robotics & Automation (ICRA). Hong Kong.
Gerla, G., 2005. Fuzzy logic programming and fuzzy control. Stud. Logica 79 (2), 231–254.
Hamrita, T.K., Tollner, E.W., Schafer, R.L., 2000. Toward fulfilling the robotic farming vision: advances in sensors and controllers for agricultural applications. IEEE Trans. Ind. Appl. 36 (4), 1026–1032.
Hough Transformation. (2016). (The MathWorks, Inc.). Retrieved from: http://www.mathworks.com/help/images/ref/hough.html.
Husky Unmanned Ground Vehicle Robot, (n.d.). (Clearpath Robotics). Retrieved from: http://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/.
Introduction: PID Controller Design, (n.d.). (University of Michigan). Retrieved from: http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID.
Jackal Small Unmanned Ground Vehicle, (n.d.). (Clearpath Robotics). Retrieved from: http://www.clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/.
Lefcourt, A., Kistler, R., Gadsden, S., Kim, M., 2016. Automated cart with VIS/NIR hyperspectral reflectance and fluorescence imaging capabilities. Appl. Sci. 7 (1).
Mousazadeh, H., 2013. A technical review on navigation systems of agricultural autonomous off-road vehicles. J. Terramechanics 50 (3), 211–232.
Patel, P., 2015, May 6. Cheap Centimeter-precision GPS for Cars, Drones, Virtual Reality. (IEEE Spectrum) Retrieved March 15, 2016, from: http://spectrum.ieee.org/tech-talk/transportation/self-driving/cheap-centimeterprecision-gps-for-cars-and-drones.
Rejas, J.-I., Sanchez, A., Glez-de-Rivera, G., Prieto, M., Garrido, J., 2015. Environment mapping using a 3D laser scanner for unmanned ground vehicles. Microprocess. Microsyst. 39 (8), 939–949.
Rows of Lettuce, (n.d.). (National geographic Partners, LLC.). Retrieved from: http://intelligenttravel.nationalgeographic.com/2012/06/12/the-lettuce-of-wrath/salinas-ca-rows-of-lettuce/.
Sailan, K., Kuhnert, K.D., Karelia, H., 2014. Modeling, design and implement of steering fuzzy PID control system for DORIS robot. Int. J. Comp. Commun. Eng. 3 (1), 57–62.
Sistler, F.E., 1987. Robotics and intelligent machines in agriculture. IEEE J. Robot. Automation, RA 3 (1), 3–6.
Takagaki, A., Masuda, R., Iida, M., Suguri, M., 2013. Image processing for ridge/furrow discrimination. In: 4th IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post Harvest Industry. Espoo, Finland.
Xue, J., Xu, L., 2010. Autonomous agricultural robot and its row guidance. In: 2010 International Conference on Measuring Technology and Mechatronics Automation, pp. 725–729 (Changsha, China).
Xue, J., Zhang, L., Grift, T.E., 2012. Variable field-of-view machine vision based row guidance of an agricultural robot. Comput. Electron. Agric. 84, 85–91.
Yang, J., Dang, R., Luo, T., Liu, J., 2015. The development status and trends of unmanned ground vehicle control system. In: 2015 IEEE International Conference on Cyber-technology in Automation, Control and Intelligent Systems (CYBER). Shenyang.