Autonomous Ground Vehicle Navigation in Orchards Using Aerial Data Support

Kartik Gupta, Bill Mania, and Vason P. Srini

July 2015

Berkeley, CA

Abstract

Autonomous ground vehicles (AGVs) have the capability to support the monitoring of trees for yield management, disrupting critters such as squirrels and rabbits from the orchard, mowing, sweeping, and spraying. Depending on the growth of trees AGVs might be in GPS denied situation in the windrows of the orchard and thus making waypoint navigation difficult. We propose an approach for traversing the windrows of the orchard using aerial images, GIS maps, and used specified constraints. We first process the aerial image to identify the windrows of the orchard. We then construct a travel plan for the orchard to visit all the trees taking into account user constraints. Path planning is then developed and optimized to reduce repeated traversal of windrows, total distance travelled, and total time taken.

1. Introduction

The agricultural environment offers a very different set of circumstances from that encountered by a laboratory mobile robot. In one respect, operation is simplified by the absence of clutter typically present in the indoor environment; however, a number of additional complications are raised. For example, the operating areas are large; ground surfaces may be uneven; depending on the operation, and wheel slippage may be far from negligible. Cultivation may interfere with underground cables, colors may change with plant growth, and soil quality may vary. Environmental conditions (rain, fog, dust, etc.) may affect sensor function; moreover, a low-cost system is required. These disadvantages make it more difficult to realize agricultural automation. Compared with these complicating factors, agricultural farm fields have several advantages for developing autonomous guidance systems. For example, the working areas generally do not change; landmarks can be easily set up around the corners of a field and be taken as a stationary environment. The crops are always the same plants at the same places and can be easily distinguished. Therefore, even though there are more disadvantages than advantages for realizing agricultural vehicle autonomous guidance, there are enough research achievements to promote its development.

Efficient production of fruits and nuts is important for feeding the growing population of the world. Cultivating fruit trees such as peaches, apples, mangos, pears; and nut trees such as almonds, walnuts, cashews, pecans, and pistachios can benefit quite a bit from autonomous ground vehicles (AGVs). The mowing of orchards to keep weeds and grass under control, spraying with herbicide, sweeping, and monitoring the health of trees using videos can be done using AGVs. Pest control can also be done using AGVs. For example patrolling orchards using AGVs with blinking lights and sound can disrupt pests such as squirrels, rabbits, and mice. Operating agricultural equipment accurately can be difficult, tedious, or even hazardous. Automatic control offers many potential advantages over human control. Autonomous guidance of agricultural vehicles is not a new idea, however, previous attempts to control agricultural vehicles have been largely unsuccessful due to sensor limitations. Some control systems require cumbersome auxiliary guidance mechanisms in or around the field while others rely on a camera system requiring clear daytime weather and field markers that can be deciphered by visual pattern recognition With the advent of affordable GPS receivers, engineers now have a low-cost sensor suitable for vehicle navigation and control. However, the difficulty still remains with defining the path to be followed around the orchard, with minimal human intervention. This study suggests one such method for identification of the rows of trees in an orchard.

2. Aerial Image analysis

Autonomous navigation of a ground vehicle in an orchard requires a detailed map of the orchard for traversing the rows of the orchard. Constructing the orchard map from aerial photos available from Google earth or similar services is described in this section. In our project we have used the image obtained using Google's Static Map API. The image was obtained by marking out the boundaries from the larger image generated using the Static Map API.

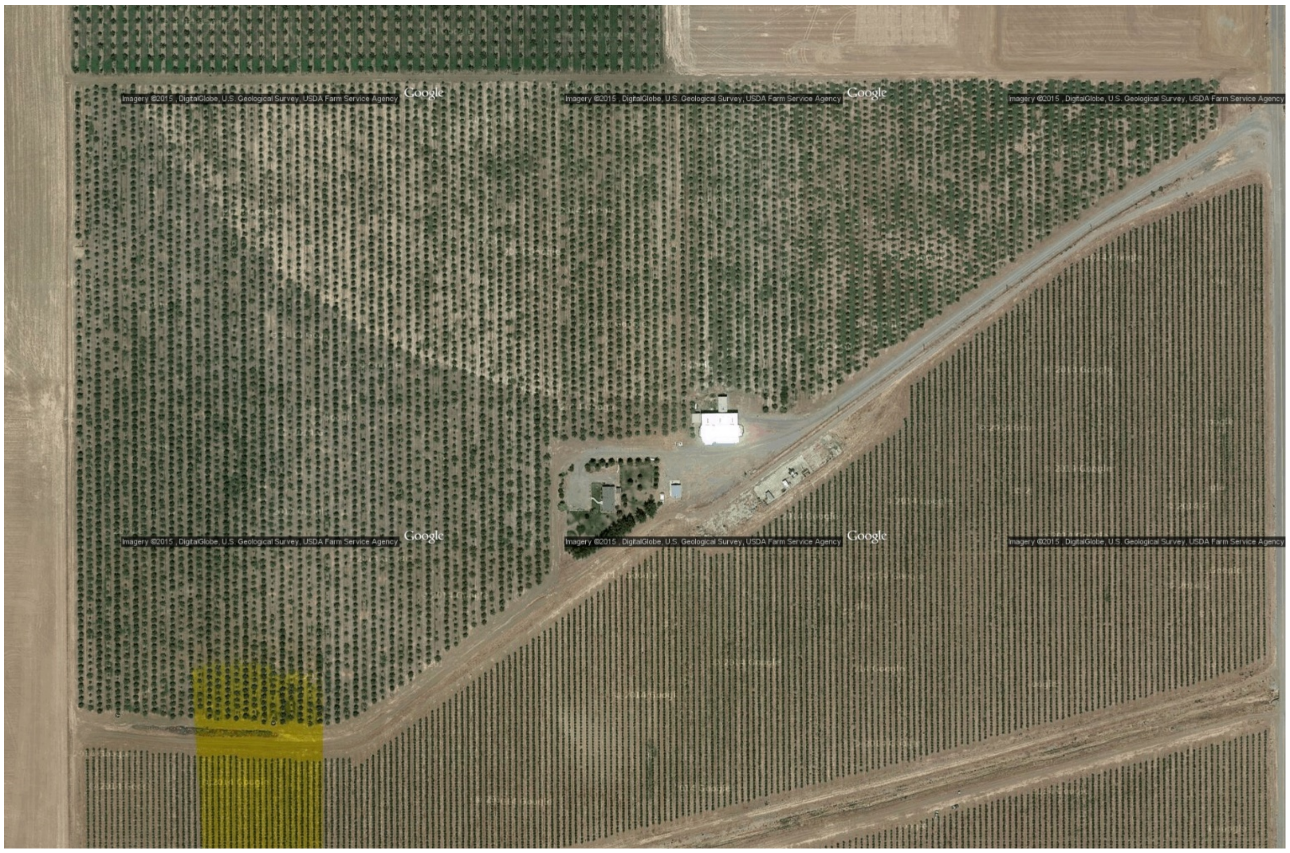

Figure 1: Aerial photo of an almond orchard obtained using Google Static Map API.

The placemarks specify the boundary of the orchard.

The initial set of constraints are the following:

1. The AGV has to stay within the specified perimeter.
2. It also has to avoid collision with trees and stationary objects in the orchard.
3. In this case the rows of trees are 20’ apart. In each row the average distance between trees is 14’. The trees are planted on the berm in a row and the berm cannot be traversed. It is an area to stay away. The robot can only travel along the rows that are 20’ apart.

It is a good idea for the robot to stay in the middle of the row.

The steps to be performed on the image using the constraints is listed below.

1. Identify the windrows of trees in the image.
2. Identify the end of the rows and the major waypoints of the path.
3. Identify how to turn from one row to the adjacent row on the left or right.
4. Develop a strategy to cover all the rows in the orchard.
5. Generate a list of all the rows and turns forming a detailed map (orchard row map).

The approaches developed for each of the five steps is explained in this section.

1. <u>Identify the windrows of trees in the image.</u>

The process of identifying windrows of trees from the aerial photograph has been completed using two approaches – color gradient and template matching. While both the color gradient approach and the template matching allows us to produce near perfect results, each have their limitations. We describe both approaches. The row identification task consists of two major sub-tasks:

1) identification of the trees
2) identification of the windrows using the identified trees

A: <u>Using Color Gradient</u>
For the first part of the task, a few approaches have been tried.

a) We started with the Sobel's algorithm and the Laplace parameter for identification of edges of the trees. These methods did give good edge detection results. However for the subsequent task of organizing them into rows, finding the center of the trees was important, which required finding contours. This proved to be difficult and it resulted in erroneous detections.

b) Continuing with edge detection, we used the Canny Edge Detector algorithm. Automatic thresholding selection was used based on the mean and the standard deviation of the grey scale convert of the original image. Using this method we were able to detect approximately 90% of the trees. It worked better for integrating with the contour function. While this gave very good results, three major problems were observed:

1) The contour function often detected some trees in a row as a single contour, due to trees being nearly continuous in the image. Thus using the mass center of the contour as the center point of three led to loss of information and sometimes led to erroneous results.

2) Often the same tree was detected as more than one contour. This was due to partial or double edge detection of the tree. Sometimes the same tree showed varying contrasts in the image leading to double detection.
3) The areas with dense grass/weed growth on the ground were also identified as contours. These points led to many wrong row detections.

A workaround to these problems was found by ignoring the detected contour as a tree and rather using the minimum enclosing rectangle of the contour for subsequent analysis. The line parallel to the length of the rectangle and passing through the "mass center" of the related contour of the enclosed rectangle, subsequently referred to as the "mass-central-line", was rightfully chosen as the center of the row.

This led to the following improvements in the earlier reported errors:

1) The detection of multiple trees in a row was easily integrated into the results as the enclosing rectangle accounted for all these trees.
2) The dense grass/weed growth led to big sized enclosing rectangles, which were easily filtered out. The first step of the filtering process was based on the selection of the approximate tree width by the user in GUI environment. The user was required to slide across a scroll bar which caused selection of tree widths and showed the trees that lied within that approximate limits of the width to be enclosed by boxes. This simple selection made the first step of the filtering process extremely easy to use for the user and resulted in effective results. This is selection is shown in the figure below.

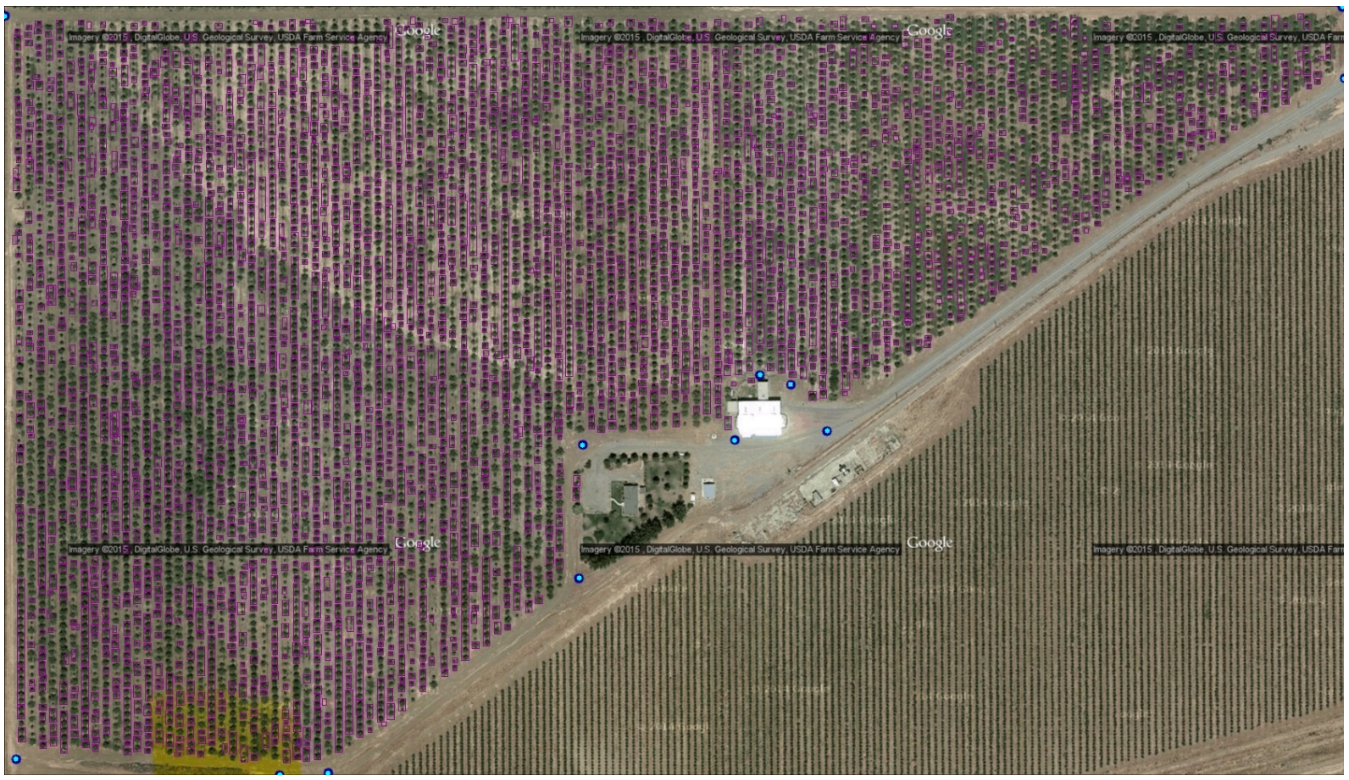

Figure 2 : Selecting approximate tree-width. The purple boxes denote the detected trees.

3) By using effective filtering techniques, the duplication of trees in the result was carefully avoided. The filtering process simply involved checking for overlapping or enclosed-within-another enclosing rectangles.

B: Using Template Matching

In the other approach, we decided to detect trees not based on color gradient and contour detection but by template matching. While the method is computationally intensive, it gives us much better results. By the very principle of this method both of the above problems mention in << to be inserted>> were sorted out. While lesser trees were detected, they were perfectly organized by the row-detection algorithm giving near perfect results. However template matching requires initial selection of the template tree manually which conflicts with our aim of making the identification process completely autonomous. Thus this approach was rejected.

2. Identify the end of the rows and the major waypoints of the path.

The next task was to identify the rows by linking correctly the identified enclosing rectangles. This was done by connecting the relevant enclosing rectangles in continuation as per the selection criteria, one row at a time. As per selection criteria first the staring rectangle for a row was selected. Next, the most suitable rectangle for the starting rectangle was found and then the process was iterated by finding the most suitable rectangle for the previously found box. The suitability of the rectangle was estimated by looking at its placement with respect to the given rectangle. It was said to be suitable if its placement and size satisfied any one of the seven cases, as shown in the figure.

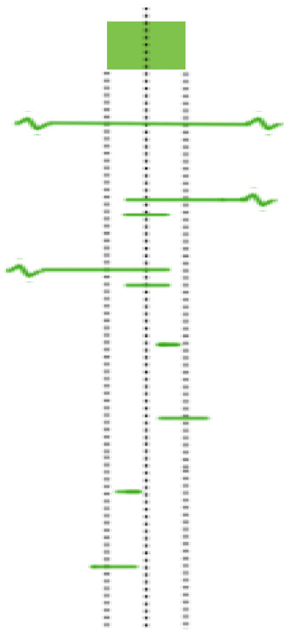

Figure 3 : The green rectangle represents the enclosing rectangle with respect to which the most suitable rectangle is being found. The green lines denote the acceptable widths and placements of the next rectangle.

After a rectangle was found suitable, the row was extended by joining the mass-central-lines and the rectangle was marked as used and ignored for suitability checking for further rows. However, if none of the unused rectangles satisfied the suitability condition then it was assumed that the row has reached its end, and the row extended till the boundary.

After the set criteria for ending a row was achieved, the search for the next row was started. The starting point of the next row was selected by considering specific distances ,in terms of the row width, from the previous row and then moving on to find the rest of the rectangles that make the row by the process mentioned above.
Often in dense orchards or low resolution photos of the orchards all the trees are not identified. This greatly influenced the selection criteria as we were often required to identify rows in cases where many trees might not be identified.

The major waypoints of the path are selected by generating points that are at a distance equal to half the width of the rows from the end of the rows, in a direction at right angles with the direction of the rows towards the right of the rows when seen in the image of the orchard.

3. Identify how to turn from one row to the adjacent row on the left or right.

A major problem is faced by AGV during turning at corners due to C2 discontinuity in the path. To solve this problem we have used Bezier curves at each of the corners. A Bézier curve is a parametric curve frequently used in computer graphics and related fields. Bézier curves are used to model smooth curves that can be scaled indefinitely. The algorithms impose constraints such that curve segments are $C2$ continuous in order to have curvature continuous for every point on the path. In addition, they give the reference path more freedom. The degree of each Bezier curve segments are determined by the minimum number of control points to satisfy imposed constraints. The optimized resulting path is obtained by computing the constrained optimization problem for the cost function described. The numerical simulation results provided in the paper demonstrate the improvement of trajectory generation in terms of smoother steering control and smaller cross track error. We have joined three 4 degree Bezier curve for making each turn around the end of the row in effect making a u-turn. This ensures greater smoothness and functioning ease for the AGV.

4. Develop a strategy to cover all the rows in the orchard.

It was chosen as a convention to start the traverse from the leftmost row and moving progressively to the right end of the farm. This required making u-turns alternatively towards the left and right at the end of each row. By this simple strategy all the rows are covered only once, providing an efficient way for traversing the orchard.

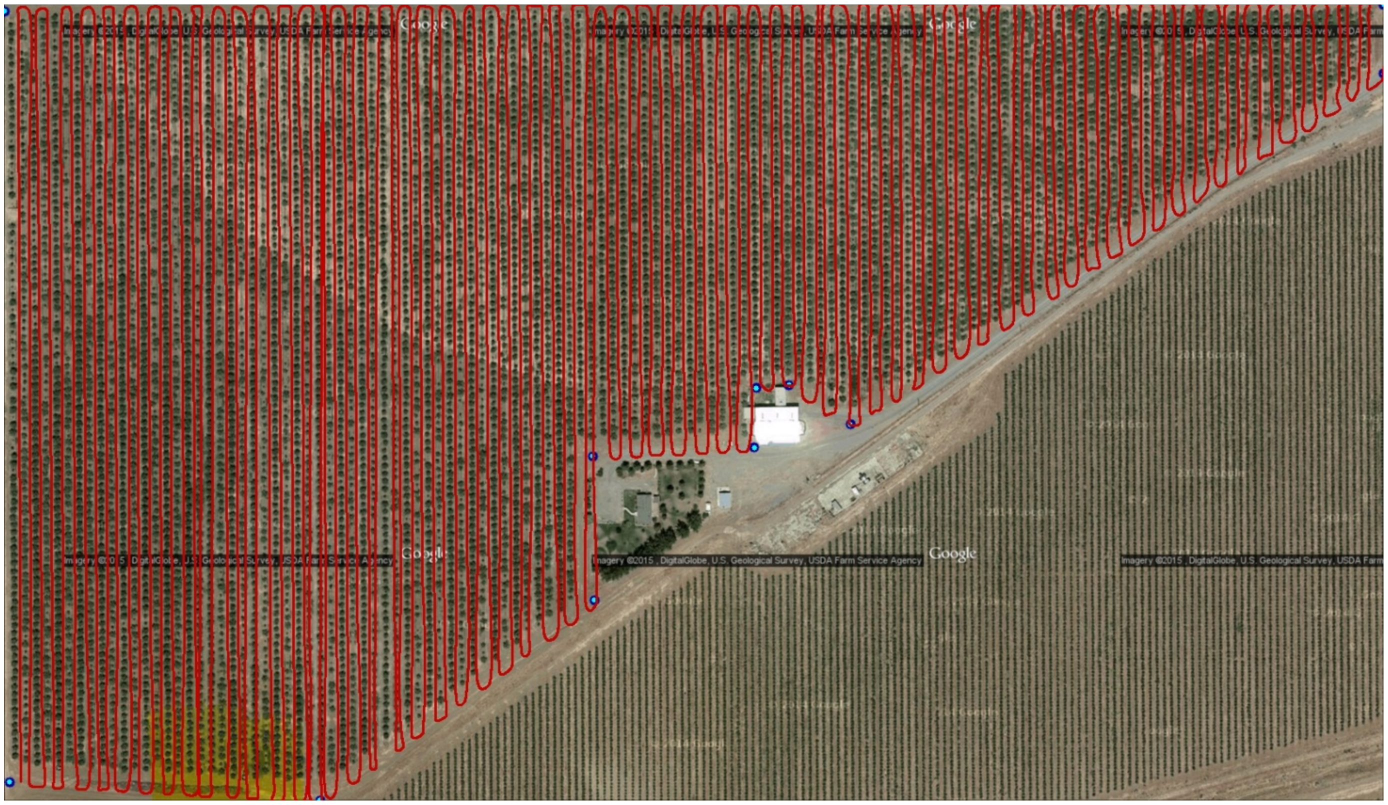
Google
Imagery ©2015, DigitalGlobe, U.S. Geological Survey, USDA Farm Service Agency

Figure 4 : Final Generated Path

3. Path Planning

Path planning involves two steps: global path planning, and detailed path planning.

The global path planning, also known as mission planning, provides the route to be taken using the orchard row map so that all the rows of the orchard will be traversed. Initially the global path planning will focus on covering all the rows. Optimizations can be done to reduce the repeated traversal of rows, time taken to complete the traversal, and improving the safety of the traversal.

The detailed path planning is concerned with the actual travel of the robot on the row, staying in the center of the row, making turns to go from one row to another row, avoiding collision with trees or objects on a row, making smooth turns, and keeping the robot in a safe state at all times. The robot must respond to emergency stops and also inform the control center if the robot cannot move due to unforeseen situations or mechanical/sensor/software problems.

4. Implementation using ROS Nodes

Two nodes have to be designed for doing autonomous navigation in an orchard. The first one is the global navigation node, called orchard_navigate, which takes as input the coordinates of the center of the orchard, generates the map using using the Google Static Map API and provides a list of rows to be traversed. This list has to be updated based on the actual traversal that was done by the robot and recomputed if needed. The second node is the detailed local navigation node, called orchard_localnav. The existing path planning nodes in ROS have to be updated with the smooth curve planning for making turns and collision avoidance with trees. Another important item is to keep away from the berms.

Robot Starting Challenges

One of the challenges is moving from a parked spot or a warehouse to start the traversal. If GPS is available then current position information can be placed on the orchard row map and the nearest row to start the traversal can be determined. Assuming that a few meters of GPS error is a possibility, correction has to be done using local 3D monument information. Constructing the 3D monument database and a detailed map of the orchard requires performing SLAM during traversals. Since the robot will be traversing the rows several times in a month strategies for building a good map and 3D monument database have to be devised.

If GPS is not available then camera and radar data have to be used.