1. Introduction

In an effort to make vehicles that demonstrated artificial intelligence (AI) abilities, engineers must design systems that allow the vehicle to correctly perceive the local environment and identify features of interest. Many autonomous vehicles (AV) use a camera to gather vast amounts of data from the environment and then analyze the data to find features that give information about the AV’s position to these features. Land based AVs are often interested in finding lines that on the ground that represent lane markers, wall edges, or a host of other things. A common technique for line detection is the Hough transform because of its ability to detect a line even when there is a lot of noise in the image and because it returns how strong a line is in an image. The Hough transform returns an angle and distance to line from a specific origin. Because computers traditionally classify images with the origin (0, 0) in the top left screen corner, this location is often used as the origin for the Hough transform as well. The rho and theta are meaningless by themselves and must be transformed back into Cartesian coordinates which then must be mapped to real world coordinates. However, by simply moving the origin of the image to the bottom center, the results of the Hough transform are immediately useful to real world coordinates without any farther mathematical manipulation.

1. Hough Transform
2. Technique

Although edge detection in an image is a relatively simple task for a computer, extracting information from an edge detection algorithm about lines in an image is non-trivial. The Hough transform’s usefulness is in its ability to identify lines in an image when it is given a set of points of interest and then used to find the line that is collinear to the largest number of points of interest.

Because two points represent a line, an infinite number of lines go through single point of interest at (x, y), and each line is normally represented in a simple form. The Hough transform seeks to find the equation of all lines separated by a small angle through every point of interest. However, computers can not represent certain lines in normal form such as vertical lines which present a problem as the slope and y-intercept go to infinity. A solution to this problem is to represent a line by a different set of parameters: its distance from a chosen origin, and by the angle to the closest point on the line. (picture1 showing y = mx+b and theta and rho). For a given point (x,y) every line passing through it can be represented using this equation.

Solve for r.

For a given point , the equation allows a computer to increment theta and compute the corresponding rho for every line passing through the point using values that do not approach infinity. When the rho-theta values are mapped into a rho-theta space, the result is a sinusoid, so that a point in Cartesian coordinates is represented by a sinusoid in rho-theta space corresponding to all the possible lines passing through the point. By converting all points of interest in an image to sinusoids in rho-theta space, and then finding where the sinusoids most frequently intersect, a rho-theta value is found that represents a line which is the most collinear of points of interest in the original image. After the rho-theta value is found, various techniques use different methods to convert the rho-theta value in to more useful information. For example, “bob” (citation, same paper about width checking) converted the rho-theta values back into y=mx+b form, pasted the line into the original image, computed the points on the image where the line goes out of the camera’s view, calculated using perspective geometry where these point corresponded to on the ground, and then stored the ground points into a map.

1. History

In 1962 Paul Hough submitted a patent for the Hough transform which in geometric terms described how a set of collinear points could be represented in a transformed space by a “knot” or point of intersection. In 1969 Azriel Rosenfeld presented a way of using a computer array to represent the transform space with counters so that many collinear points will result in high values or the bins. In 1972 Richard Duda and Peter Hart published “Use of the Hough Transformation to Detect Lines and Curves in Pictures” which gave the modern mathematical basis and implementation of the Hough transform as it is commonly thought of today. Currently the Hough transform is used in nearly all fields of computer vision analysis.

1. New Technique

Because the top left corner of a picture represents the origin of pixel locations, it is often used as the origin for the Hough transform as well. By translating pixels locations so that the bottom center of the picture is the origin the resulting rho-theta from the Hough transform can be immediately useful for a mobile robot. The rho value would relate to the distance to the closest point on a line from the robot and the theta value would represent the angle to the closest point on the line.

1. Evaluation
   1. Computation

Although moving the origin to the bottom center results in immediately useful results from the Hough transform these results are at the computational cost of translating ever pixel of interest’s x value over half the screen width and y value to be screen height minus the pixel y value.

A similar result can be found by transforming the final answer from the rho-theta from the top left corner to the rho-theta from the bottom center by this equation.

Where are the values returned from the Hough transform when it is computed around the original origin, are the coordinates of the new origin (in this case a = half the image width and b = image height, and are the rho-theta values as though they were computed around the origin at .Because Sine and Cosine are transcendental functions and take more computational power on a float point processor than simple algebraic calculations, the computational requirements can be reduced from nine transcendental functions to four by immediately finding the location that corresponds to the rho-theta at the top-left and then transforming the answer using these equations.

Because of the computational complexity of the transform when it is necessary to find many lines in single image it may be computationally more efficient to translate the pixel coordinates before calculating the rho-theta values rather than repeatedly transforming the resulting rho-theta value to represent a rho-theta at the bottom center of the image.

* 1. Singularities

Although the Hough transform robustly finds lines in a noise image, it will fail to give a correct line when rho = 0. In an effort to reduce computational requirements the rho and theta values are often divided by a factor so that the rho-theta space takes less memory and less time to search for the rho-theta bin with the largest value. As a result of dividing by a factor, the rho-theta values are quantized so that they can only be multiples of the factor and the possibility that rho= 0 is increased. This problem can be eliminated in a number of different ways. By adding a small value to the rho at the end of its calculation, the Hough transform will never have a rho = 0 value. This solution sacrifices a small amount of accuracy to ensure that rho never = 0. Another solution is to place the origin in a place where you never expect a line to go through it. The bottom center of the picture would be an ideal place for any robot that is seeking to stay between lane boundaries. The lane markers should never go through the bottom center of the picture unless the robot has already driven significantly off course and is now straddling the line.

* 1. Ease of implementation

Translating the pixel location values so that the bottom center is the origin is generally easier to implement into a simple AV because the coordinates will represent a more traditional Cartesian coordinate system where the center of image is, left is, right is, the bottom of the image is , and the max y value is the top of the image. Because it is often difficult to think through the computer coordinate system where the top left is the origin and down is positive y, translating the pixel values can reduce the time spent attempting to think through how to correctly adjust for computer coordinates.

1. Testing

Testing of the two methods took on two forms. First a simulation of the Bob Jones University IGVC robot was created. The simulation used the Microsoft XNA frame work to facilitate use of 3D graphics to simulate a virtual world and robot with a camera. XNA ‘s robust ability to simulate all sorts of things in X-Box 360 games make it ideally suited for creating simulations to test many engineering problems that involve complex graphics. By loading real pictures of grass into the simulation, the image analysis from the virtual robot’s camera ran into the same problems that the real robot has of seeing lots of noise in a picture because of shiny grass. The simulation allowed for initial testing of both methods of the Hough transform while varying many of the parameters.

Second, the algorithm to transform the coordinate system was placed on the BJU IGVC robot. By taking several measurement of the actual distance to a line from the bottom of the camera’s view while recording the rho value from the Hough transform a set of data was made. Using a simple 3rd degree polynomial fit a function was found that related the rho value to the actual distance to the line. The angle to the closest point on the line was also measured from the bottom of the robot’s viewing area.

1. Results

The simulation effectively proved that the computation time to run the new Hough was … compared to the old Hough when it was detecting 10 or more lines. (Need to work on getting more accurate data, my initial testing showed that difference in time it takes to compute each method is statistically identical. I need to develop a more accurate method of testing before presenting any data).

Testing on the BJU IGVC robot revealed that the distance to the line computed by the algorithm was within X error when the line was within 1 meter and y error when the line was greater than 1 meter. At any angle the algorithm’s theta value was consistently accurate to +-5 degrees.

1. Conclusion

By moving the origin around which the Hough transform is calculated to the bottom center and reversing the direction of the y-values so that up is positive, the rho and theta values from the Hough transform are immediately useful data to the user about the position of the line when for AVs with a camera mounted on the front and pointed down and forward of the vehicle. The computational cost of translating every pixel to the new coordinate system using simple arithmetic may be computationally less than transforming the rho-theta values so that they are with respect to a different origin. A formula to convert from the rho-theta values returned by the Hough transform from one origin to another was found. Picking a different origin or transforming the rho-theta value of the Hough transform to be with respect to another coordinate system is an easy way to find useful data that directly correlates to real world values without the need for post Hough transform calculations.

 