Virtual Walls for Lewis

By: Trisha Leconte

Santa Clara University
September 2006

Virtual Walls for Lewis

1. Abstract

This paper presents research in robotics focusing on constructing and recognizing virtual walls. Washington University has a robot named Lewis who is about 5 feet tall and weights about 500 pounds. This paper describes the part of Lewis' framework used to implement virtual wall service. It describes the mathematical procedure to find the intersection of a virtual wall and laser.

2. Introduction

Washington University is currently doing research in robotics. The robot is equipped with lasers to acknowledge its position and as well as objects in 3D space. Over the last few months, Lewis' framework was rebuilt from scratch. One of the modules that had not been created was the ability to alter laser readings to recognize virtual walls. As Lewis moves around in space, he is constantly updating and taking 360 laser readings over a span of 180 degrees from left to right in front of him. The laser readings are conveniently stored in a vector with 361 indices. Each numerical value in the vector is the distance from the robot to the object or virtual wall.

3. Background

My project is titled Fenced Laser Service. The service allows virtual walls to be defined in the real world. This service also allows a user to define areas in which the robot is allowed to operate and to treat the boundaries of these areas just like physical walls in the environment.

There are two frames of reference that need to be specified when talking about coordinates. The first is absolute/world reference, which entails the coordinates in the environment. The second is relative coordinates, which entails the coordinates' position relative to the robot. In relative coordinates, the robot is always located at position (0,0).

A virtual wall is defined by two x and y coordinate points. The coordinates, when first defined, are in the world reference frame. For simplicity it was chosen to define them as simple line segments instead of complicated paths.

4. Mathematical Procedure

A few linear algebraic equations are used to find the point of intersection of two lines. Several cases were taken into consideration when determining if there is an intersection between the two line segments, not just between the two continuous lines. If an intersection point is not found, the corresponding vector value does not change. If an intersection is found, then the Pythagorean Theorem is applied to find the distance from the robot to the intersection point and then that value is written over the corresponding vector value. This process of finding the distance from the robot to the intersection of line segments is repeated 361 times from index 0 to 360, or PI radians.

Before beginning the algorithm to find the intersection of two line segments, it is imperative that the robot understands the definition of a virtual wall. To define an instance of a wall in C, a class called Wall was written. It consisted of two coordinate points describing the line segment of a wall and a character string for the name of the wall.

The laser readings from the robot come in the form of a vector, having a magnitude and a direction. There are 361 indices all filled with distance values. To find an angle corresponding to an index a simple formula was created. The laser spans PI radians and there are 361 indices, so the angle per index would be PI radians divided by 361 indices. The angle for any given index would be the angle per index multiplied by the vector index of choice.

With the angle, the vector point of the laser reading can be found. The x component can be found by multiplying the vector distance by the cosine of the angle. The y component can be found by multiplying the vector distance by the sine of the angle.

By this point, the module now has four recognized points. The line segment coming from the vector has points (0,0) and (D*cos(angle), D*sin(angle)) where D is the vector distance. The two points defining a wall segment are read from a list of walls that are read by the module. There are several cases written to determine if a laser reading intersects a virtual wall.

Before entering the test cases, the module first distinguishes the leftmost from the rightmost points of the virtual wall as well as the laser vector. For vertical lines, the leftmost point is considered the top point and right rightmost point is considered the bottom point. The purpose of doing that is to be able to apply the comparative operators to two left or right points.

The first case deals with vertical laser readings and virtual walls. If the x values are the same that means that the two segments lie on the same line. If the y value of the laser reading is in between the y values of the wall, then altered laser vector would be the edge of the wall closest to the robot. If the y values of the laser vector fall short of the wall, then the distance in the vector does not change. If the segments are found to intersect, then the altered length would be the absolute value of the y value of the intersection point.

The second case deals with a vertical wall and any non-vertical laser vector. The x and y intercepts are found by using the intercept formula, y=mx+b, with b=0. If the y intercept is in between the wall coordinates and the x intercept is in between the laser vector coordinates, then the segments intersect. The Pythagorean Theorem is applied to the y and y intercept points to find the new altered laser reading distance.

The third cases deals with any non-vertical wall and non-vertical laser vector. First, the x intercept is found by dividing the wall intercept (y=mx+b) by the wall slope multiplied by the leftmost x wall coordinate. The y intercept is found by multiplying the vector slope and the x intercept. The wall angle is found by taking the arc tan of the wall coordinate points. If the x intercept is between the two x values of the laser vector and the x intercept is between the two x values of the wall, then two segments intersect. The walls are parallel if the wall angle and the vector angle are the same.

If they are the same, there are a couple of cases to check to see if they intersect. First test case checks to see if the right or left points of the laser vector lies on the wall segment. If it does, then the Pythagorean Theorem is applied to the point of intersection closest to the robot. The second case checks to see if both points of the wall lie within the laser vector. If true, the point closest to the robot is used in the Pythagorean Theorem and that distance is returned.

5. Conclusion

As the robot moves around in the real world Lewis is continually collecting laser data in relative coordinates. The algorithm to find intersections and return a new laser vector is looped and the laser vector is continually updated. For every index value, these cases are repeated for every virtual wall that is contained in a list of walls. Each time, the laser vector of data is updated. The Pythagorean theorem as well as the slope intersection formulas are used to find the intersection point of the laser and the virtual wall.

6. References

Hauke, Humanoid. Multi-Cue Localization for Soccer Playing. University of Freiburg. 11 Nov. 2007. <http://citeseer.ist.psu.edu/rd/87265519%2C762123%2C1%2C0%2CDownload/http%3AqSqqSqwww.informatik.uni-freiburg.deqSqhrqSqpapersqSqRS06_Strasdat.pdf

Dellaert, F. and Fox, D. and Burgard, W. and Thrun, S. Monte Carlo Localization for Mobile Robots. Carnegie Mellon University. 11 Nov. 2007. <http://citeseer.ist.psu.edu/rd/87265519%2C35242%2C1%2C0%2CDownload/http%3AqSqqSqwww.cs.cmu.eduqSqPeopleqSqdellaertqSqftpqSqicra99.ps.gz>