Approaches to Pole Detection using Ranged Laser Data

Shaun Press[1] **David Austin**[1,2]
shaun.press@anu.edu.au d.austin@computer.org

[1]Robotic Systems Lab, RSISE
Australian National University,
ACT 0200, Australia

[2]National ICT Australia,
Locked Bag 8001,
Canberra, ACT 2601

Abstract

In building up maps for mobile robot navigation the detection of clearly identifiable landmarks is important. Also important is the confidence level that recognised landmarks are really what we identify them as. This paper looks at using laser range sensing to both identify poles and to attach a probability to poles observed. Adapting the standard method of circle detection using Hough Tables a new approach to assigning probabilities to features is proposed. These probabilities are assigned according to the level of support in the Hough Table and according to aspects germane to the objects itself. Results are given demonstrating the real time application of this approach and its usefulness in mapping landmarks.

1 Introduction

Self navigation by mobile robots relies on being to clearly identify landmarks within their environment. By identifying significant features Robots can not only generate more efficient traversal paths, but more importantly, can also use the landmark information to assist in localisation [Thrun *et al.*, 1998] [Kouzoubov and Austin, 2004]. To do so they must first collect data from the surrounding environment, and then apply pattern matching techniques or algorithms to clearly delineate the features of the environment.

One such feature that is especially common with modern buildings is the pole[1]. The pole could be anything from a structural support to the leg of a table. In the case of permanent poles (such as building supports), the existence of poles can also provide additional navigational information, due to their regular spacing.

This paper looks at the various techniques for identifying poles, with particular emphasis on using Hough counts to assign probabilities to potential poles within the environment.

[1]A pole is any curved surface, not necessarily forming a complete circle

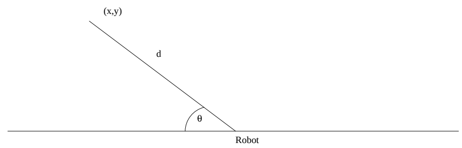

Figure 1: Calculating x and y relative to the robot. $x = dcos\theta$ $y = dsin\theta$

2 Collecting Data

A mobile robot mounted laser scanner returns range data based on the return time of a laser beam. The laser scanner sweeps out an arc of 180° at half degree intervals. The returned data is in the format of 361 distance values which is the distance in metres of the reflection point for each laser beam. This information is then converted to Cartesian co-ordinates, relative to the robot's position [Figure 1]. This set of readings is then processed to find the existence of poles in the environment. The laser scanner returns 3 scans per second.

3 Processing Data

3.1 Finding Circles

Any three non-linear points define a circle in two-dimensional space. Therefore, by taking any three points in the data set we can identify a circle with a centre x,y and a radius r by using the perpendicular bisector method.

The Hough table is a two dimensional array which is mapped directly to the Cartesian co-ordinate space, so each circle we identify with a centre (x,y) is given a vote in the corresponding cell (x,y) in the Hough table [Atherton and Kerbyson, 1999].

Now while any non-linear 3 point set will describe a circle, it is obvious that the majority of these circles will not correspond to real circles in the robot's environment. The assumption is that real circles will contain multiple

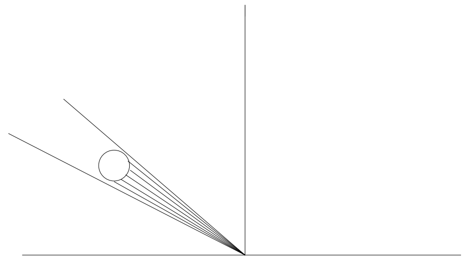

Figure 2: A subset of a laser scan striking a pole

sets of points centred around (x,y) resulting in a Hough table entry of a large number of votes [Figure 2].

3.2 Selecting Points

Assumption 1 *All points on a pole are consecutive*

The first approach to selecting 3 point sets is to assume that all circle features will be defined by a consecutive set of points. Therefore we need to only make one pass through the data set, identifying circles based on points *n, n+1, and n+2.* While this approach has the advantage of taking less processing time it is susceptible to negative effects caused by noisy data. This approach can also fail when the poles being detected are partially obstructed by other objects in the environment, as the effect of missing votes is magnified over the smaller data set.

Assumption 2 *All points on a pole are located close together*

A more robust approach is to assume the points lying on a circle are located close together. This requires more passes through the data as points that lie within range may not be stored contiguously. However this does deal with noisy data in a better way, and also provides a greater level of granularity when calculating probability values for each possible pole. For this paper this will be the approach used.

3.3 Collating Information

Assumption 3 *All circles centred on the same point are the same circle in reality*

Having identified a circle centred *x,y* we also identify the radius of the circle. When an entry into the Hough table at *x,y* is made we also need to store the radius information as well. During the processing phase the radii is summed, so at the completion a total radius of each circle centred on *x,y* is maintained. The estimated size of the circle will be the total radii divided by the number of votes. To avoid the effects of either a single erroneous reading distorting the result, the standard deviation of the radii is also calculated and used when calculating the poles probability.

3.4 Aggregating Information

Due to the discrete nature of the Hough Table indexing, there will be occasions when the centre of a circle lies on the boundary between two possible cells. This is especially noticeable when two "close" circles receive a significant number of Hough votes. By aggregating these circles together we get a more correct result for a potential pole.

The approach used is to move through the Hough table, starting with the pole with the maximum votes, and combining all poles centred within a certain threshold. This approach assumes the pole with the maximum votes is the correct pole, while other close poles are "echos" caused by noise or other external effects.

When combining poles, a weighted average approach is used to calculate the new centre and radius. The total votes for each of the poles is also combined. This process is repeated across the entire Hough space, in descending vote order.

To make this process as accurate as possible, the distance threshold between centres has to be carefully chosen. If the value is too small then close poles will not get aggregated. If the threshold is too large then the following problem occurs.

As poles are aggregated, the centre and radius of the new pole is usually different from the measured poles. This can cause more poles to be included in the aggregation process, creating a expanding wave effect, whereby poles that were not initially near, get eaten up by this roaming pole. If the distance threshold is not constant, but instead set to the radius of the aggregated pole, then this will have an adverse effect as well[Barnes and Liu, 2002].

While the problem wasn't significant for true poles, it did create a number of false positives for circles which were detected along straight but irregular surfaces.

4 Assigning Probabilities

Once all the data has been processed, each circle needs to be assigned a probability that it is indeed a real pole. This probability is made up of a number of factors including number of votes, distance of circle from robot, length of circle arc etc

The probability weightings for each of the pole features was based on a number of experiments conducted on the XR4000 simulator that is part of Dave's Robot Operating System (DROS) (http://dros.org). While these weightings will be refined as a result of future experimentation, they did generate reasonable values for the initial practical experiments carried out.

Each pole is initially assigned a probability(p) of 1.0. Then this value is down weighted depending upon the following criteria.

4.1 Length of Arc

The first criteria is the length of arc. This is determined by the number of laser points that make up the circle. Each circle has a start point and an end point, which refer to the first laser reading found for this circle, and the last laser reading for this circle.

For a perfectly circular object, with no obstructions between it and the laser, a full arc of 180 degrees will always be visible. Therefore the number of points between the first and last visible point can be calculated using the following formula

$points = \frac{2*radius}{\sin^{-1}(0.5)\sqrt{x^2+y^2}}$

The term $\sin^{-1}(0.5)$ is the distance between two points located 1 metre from the scanner. By multiplying it by the actual distance the centre is from the scanner, we get the actual distance between two points across the circles diameter. Dividing the diameter by this value gives us the number of points needed to cover an arc of 180 degrees.

Having calculated the number of laser points for a circle of radius r we then compare it with the number of points between the start and end points from our measurements.

If the number of recorded points exceeds the number of predicted points by a factor greater than 2 we reduce the probability that this is a real pole to almost 0. If the number of points an exceeds the predicted number pn by less than this, then the probability becomes

$p = p' * \frac{an-pn}{pn}$

We then apply the same process to determine whether the circle has an arc less than 90°. The number of points in a 90° arc is

$an = \frac{radius}{\sin^{-1}(0.5)\sqrt{x^2+y^2}}$

If the number of observed points is less than half of this (ie the arc is less than 45°) the probability is reduced to almost 0. Otherwise the probability becomes

$p = p' * \frac{pn-an}{pn}$

Then, for any distance over 4m we downgrade the probability to account for an increased chance of sensor error. The formula is

$p = p' * (1 - \frac{dist-4}{12})$

Finally, the deviation of the observed radii for each pole that contributed a vote for center x,y is checked. If the deviation is greater than a set threshold the probability is modified with:

$p = p' * 1 - \frac{average_deviation}{r}$ The new probability is then returned to be used as an input for the next stage of the processing.

4.2 Voting level

The major criteria for finding poles is the number of votes in the Hough table. There is both a hard minimum and a relative minimum used in setting the probability.

The hard minimum is set at 200 votes. This is derived from both experimentation and calculation. Experimental runs have shown that 200 votes seems to be right. The theoretical basis for this number is as follows. Take n as the maximum number of observed points between the first and last scanned points on the pole. $n = (l - f) + 1$ If each set of three points (P_i, P_j, P_k where $i < j < k$) contributes a vote then the maximum number of votes is

$votes = \frac{(n-2)*(n-1)*(n)}{6}$

Solving for where $votes = 200$ we get n = 11.66.

So by setting the minimum number of votes to 200 we are requiring at least 12 points be scanned for each potential pole. Any pole that receives less than 200 votes has its probability value scaled down by the ratio of votes received to 200.

The number of votes received is also compared to the number of votes that the maximum scoring pole receives. If this value is less than the maximum votes times a preset threshold then the probability is scaled down by the ratio between the two values.

5 Collating multiple observations

The above process is carried out on a single set of laser scanned data. As the robot moves around its environment it will scan for poles many times per second. These multiple scans are then used to construct a probability map of the environment.

Each scan returns a set of poles with radius r, centred on x,y and a probability p. Each time the robot moves further than a threshold distance d, the poles scanned are added to the set. When the robot has finished its mapping run, this set of poles can be collated.

If the threshold d is reasonably small, then the set will contain multiple observations of the same pole. As the pole centre x,y is based on real-world co-ordinates, and therefore dependent upon the accuracy of the robot knowing its own location, then these centre may be askew for the same pole. Therefore when processing the elements in the set of poles, all poles centred within a specified bounds are considered the same pole.

As each pole recording is processed it is either an observation of a previously found pole, or a new pole. If it matches a previous observation then the poles centre,radius and probability are recalculated using a running average and the number of recorded observations for that pole are increased by 1.

For a new pole the centre, radius and probability are the values for that observation and the observation count starts at 1.

When all the pole observations are processed the final probabilities for each pole is then calculated. This final probability is a function of the average probability of the observed poles, and the number of observations for that

Figure 3: The testing environment

pole. The higher the number of observations the higher the confidence we have in our probability value. The function chosen is a sine function, designed to discount poles with few probabilities but to quickly reward poles with many observations. $p = avgP * sin(\frac{obs*\pi}{18})$

This function generated useful results both in simulation runs and in practical experiments but future experimentation may result in a stronger function, either through the additional of new parameters, or via a more suitable function entirely.

6 Experimental Results

6.1 Experimental Setup

The Sick laser scanner is mounted on the Nomad XR4000 at a height of approximately 60cm. The test environment was the Robotics Lab and corridors in the RSISE Building at the ANU [Figure 3]. The robot was manually controlled.

The target poles in the test environment were part of the support structure for the RSISE building. They were spaced 6 metres apart (from centre to centre) in both along and across the building. There were also a number of other objects that could also be classified as "poles", such as pot plants, chairs, other robots and people.

The laser data was processed by the pole detection software and logged by a separate piece of software. The threshold probability for a pole was able to be set in the range 0.0-1.0. For the first run it was set at 0.1 (ie any pole with a probability value above 0.1 was logged), but for later runs was set to 0.5. The actual position of the pole was dependent on both the relative position of the poles centre to the XR4000 (as returned by the pole detector) and by the actual location of the XR4000 (as determined by the localiser software).

A number of data collection runs were carried out. The accuracy of the results relied upon the accuracy of the laser scanner and the localisation software. The accuracy of the laser scanner was high with a maximum

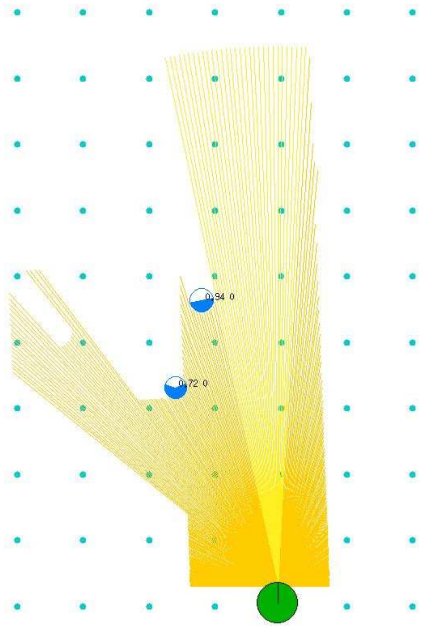

Figure 4: A laser scan of the same corridor in Figure 3

error of 2cm over an 8m distance. The ability of the XR4000 to 'know' its own position was more variable. Inexpert operation of the XR4000 would cause it to become 'lost' on more than one occasion.

6.2 Results

The XR4000 was moved slowly from its starting position inside the robotics lab along the corridor. The path chosen was a loop around half the building, returning to the starting position. Choosing this path also allowed us to see if the robot would localise correctly. For the data presented in this paper the robot did localise correctly throughout. Data collection runs where the robot became lost, and there were a number of these, were discarded.

The first run consisted of setting a low probability threshold with constant logging: Figure 5 shows the poles that were detected with a probability setting of 0.1 with constant logging enabled. Most of these poles are caused by signal noise. The occupancy grid [Figure 6] shows the number of observations for each pole, with the actual poles appearing darker than the false poles. The raw data was then passed through the mapping program and only poles with a modified probability of 0.7 or above remain. The result is shown in Figure 7.

This threshold has eliminated all the false poles created by signal noise without removing the real poles. However due to the nature of the test environment there are a few extra "poles". These include swivel chairs located in offices, a pot plant located in an open area in

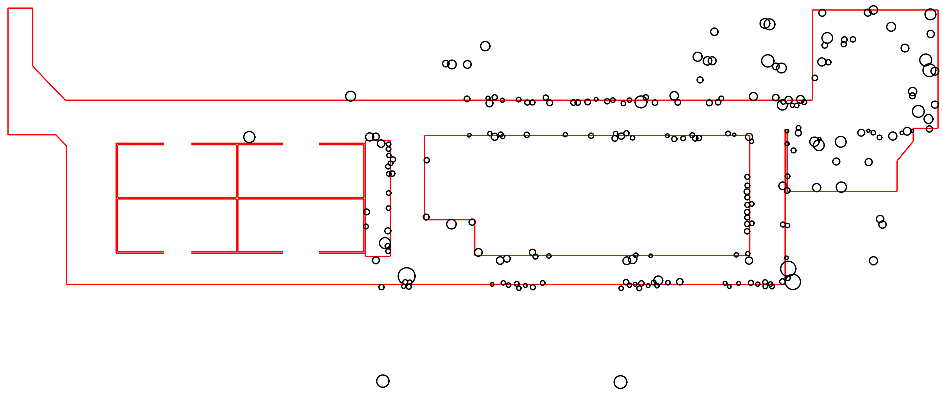

Figure 5: All poles with a probability of 0.1 or greater

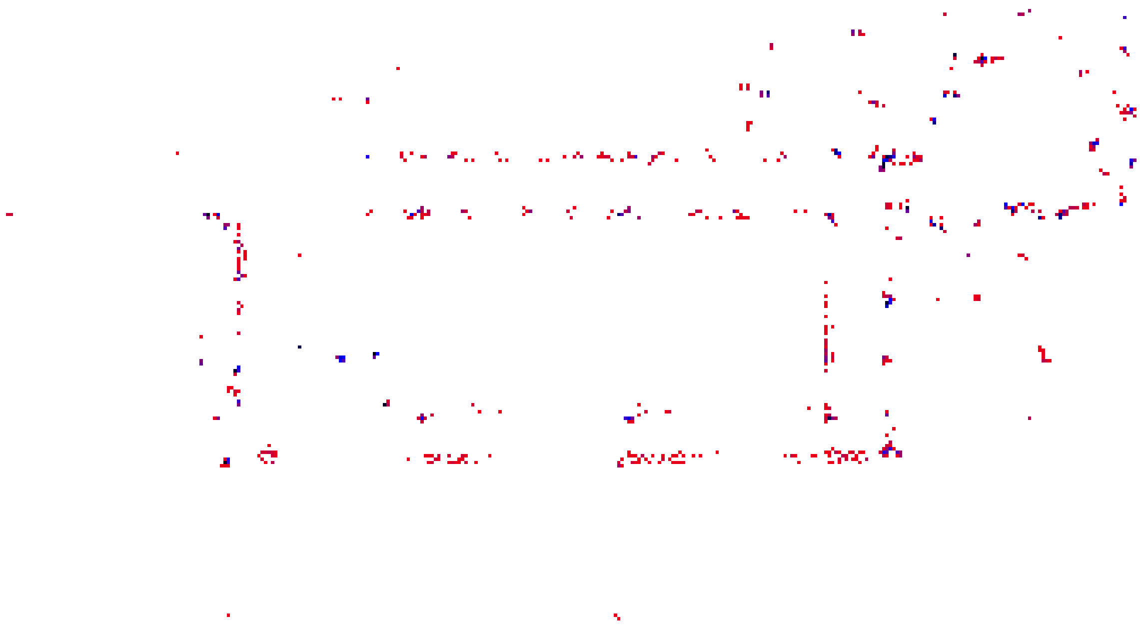

Figure 6: An occupancy grid of the same data, showing all poles found. Number of observations indicated by colour (Red low, Blue high)

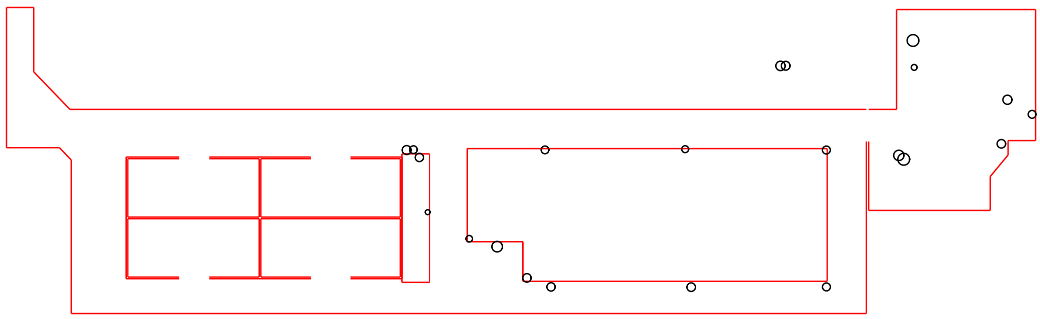

Figure 7: The remaining poles when the modified probability threshold was adjusted to 0.7

the centre of the diagram, and another robot sitting idle in the lab. Another feature of the test environment is the the poles at the top of the diagram are embedded into the corridor wall, while those at the bottom are not.

The second run had the probability threshold set to 0.5 and logging data only collected if the XR4000 had moved 0.1m in any direction. This removed the problem of some areas being logged more heavily than others due to operator effects.The path was also longer, this time consisting of an entire circuit of the building.

The data collected on this run was quite noisy, with a large number of false positives identified along the bottom wall. However these poles were only observed once and subsequent filtering eliminated them from the map [Figure 8].

The data table [Table 1] shows the best 15 poles, all with an modified probability of 0.8 or above. The real poles in the environment had a radius of 0.2 metres. They were spaced 6 metres apart with the nearest pole to the staring location (0,0) being located 1.7 metres away in both the x and y directions.

A number of corners are identified as “poles” by this algorithm. This is due to concave corners resembling curves. This in itself is not a bad thing as, like poles, they are repeatable architectural features of the environment.

7 Conclusions

The aim of this method is to both identify features in a robots environment but to also attach a confidence weighting to the features found. In correctly identifying poles in real time in a natural environment, the algorithm presented is quite accurate. It attaches a high probability to the 'real' poles. It also effectively discounts the likelihood of poles detected in noisy data from being actual poles. This probabilistic approach is at its strongest for objects located between 2m and 6m, with 4m being optimal. It also has the added bonus of being able to detect approaching humans within this range, as the human body has a shape consistent with the detection software.

Due to the robust nature of the circle fitting algorithm, there are situations where false poles are detected. The cause of this is often one of positioning and scanning the same environment from a different location and angle eliminates these results. Further experiments are being conducted to modify the probability parameters to reduce these occurrences even further.

The current application for this algorithm is in the construction of feature maps for use by mobile robots. Combined with other ranged laser feature detectors such as corner, wall and door identifiers, this approach is proving quite successful in developing accurate and useful robot maps.

8 Future Work

At this stage of the project a number of parameters and formulas have been based on results generated via the DROS simulator. Future developments will be based on further practical experimentation to optimise the parameters used for pole detection and assigning probabilities.

As the aim of this work is to provide better landmark recognition for use in SLAM [Dissanayake *et al.*, 2001], integration of this system into a larger SLAM based framework is the logical next step.

$probability$	$centre_x$	$centre_y$	$radius$	remarks
0.90	12.37	18.07	0.27	Chair in office
0.87	1.73	14.07	0.16	Real pole
0.86	5.69	29.4	0.22	Chair in cubicle
0.85	7.76	25.71	0.18	Real Pole
0.85	-0.07	6.69	0.13	Corner
0.85	1.93	35.79	0.23	Corner
0.84	2.29	34.91	0.13	Corner
0.83	1.74	20.09	0.18	Real Pole
0.83	4.58	35.35	0.25	Corner
0.82	2.31	35.38	0.22	Corner
0.82	1.77	1.75	0.17	Real Pole
0.82	1.75	32.22	0.2	Real Pole
0.81	4.55	34.95	0.21	Corner
0.80	1.75	7.83	0.17	Real Pole
0.80	7.75	19.72	0.18	Real Pole

Table 1: Second data collection run: Highest probability poles in the environment

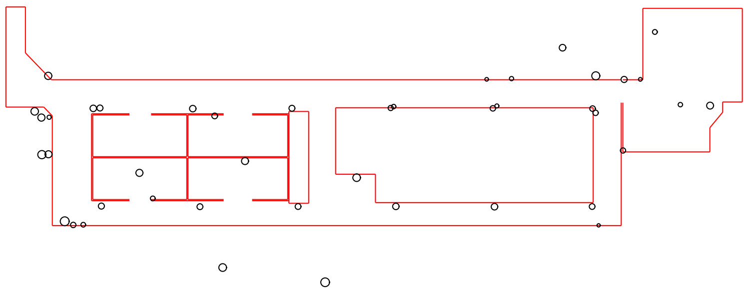

Figure 8: Second data collection run: All poles with a modified probability of 0.7 or greater

Acknowledgements

This work was supported by funding from National ICT Australia. National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program.

Technical assistance and advice for this paper was also received from Anton Chertok, Philip Blackwell, Gary Overett and Nick Barnes.

References

[Atherton and Kerbyson, 1999] T. Atherton and D. Kerbyson. Size invariant circle detection. *Image and Vision Compution*, 17:795–803, 1999.

[Barnes and Liu, 2002] N. Barnes and Z. Liu. *Knowledge-Based Vision-Guided Robots*, chapter 5. Edge Segmentation and Mapping. Physica-Verlang, 2002.

[Dissanayake *et al.*, 2001] M.W.M Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M.Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEE Transactions on Robotics and Automation*, 17(3), 2001.

[Kouzoubov and Austin, 2004] K. Kouzoubov and D. Austin. Hybrid topological/metric approach to slam. In *Proceedings of IEEE Conference on Robotics and Automation, April 2004*, 2004.

[Thrun *et al.*, 1998] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.