Fast Mapping using the Log-Hough Transformation

B. Giesler, R. Graf, R. Dillmann
Institute for Process Control and Robotics (IPR)
University of Karlsruhe
D-76128 Karlsruhe, Germany
{giesler|graf|dillmann}@ira.uka.de

C.F.R. Weiman
formerly at
HelpMate Robotics, Inc.
Shelter Rock Lane
Danbury, CT, 06810, USA
weiman@pcnet.com

Abstract: *Environment mapping is a very complex procedure that requires high CPU performance. For the last few years, laser scanners have become more and more important for mobile robots. Using their data requires many transformations between different coordinate systems. This new approach deals with a mapping within the coordinate system of the scanner, therefore it is very fast. The Log-Hough Transformation performs line finding in scans in a very efficient way, which speeds up mapping.*

1 Introduction

Mobile robot navigation depends on the knowledge of the environment. Therefore nearly all robots run a mapping task to generate or improve their maps. For map representation, usually the Cartesian coordinate system is preferred because it allows the representation of environment features independent from the robot's position. If mapping is done in an indoor environment (which is assumed for this article), the features themselves are most often represented by line segments.

In the last few years, very fast and precise laser scanners have become the most important sensor for mobile robots for all navigation tasks, such as collision avoidance, position correction and mapping. The last two problems have to be solved in the Cartesian coordinate system of the environment. This requires a powerful computer, since the data of the scanner is given in polar coordinates and each scan has to be transformed.

Present solutions transform every single point of the scan and then extract the edges. The new approach presented in this paper deals with edge extraction in the egocentric polar coordinate system of the laser scanner. Therefore only the detected edges have to be transformed, which saves a lot of CPU power.

2 Conventional line detecton

The main problem of navigation based on laser scanner data is to connect the single points obtained from the scanner to edges suitable for use by the robot. Figure 1 shows such a scan.

Several algorithms exist to extract edges from point data which has been transformed into the Cartesian system.

Line tracking. This algorithm, as described in [8], successively processes all points from the scanned set, generating a line equation. This equation is iteratively improved as points are found to lie on the postulated line. The measurement error of a point respective to the line is computed using a least squares method. Then it is tested against a certain threshold to find out if the currently examined point lies on the line. When the error is greater than the threshold, the line is terminated at that point, and a new line is started.

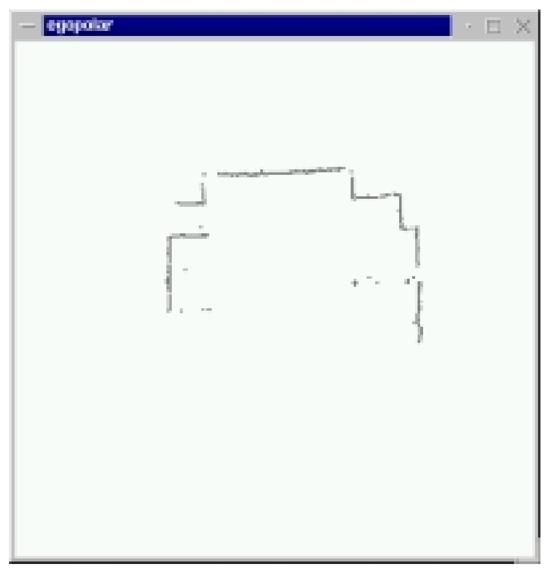

Figure 1: Single laser scan

Iterative endpoint fit. A line is generated connecting the first and last point of the set of points. The point with the greatest orthogonal distance towards the line is determined. If the distance is larger than a threshold, the line is split in two, connecting the end-points of the previous line with the newly found point. The set,too, is split in half at the point's position. This yields two sets and two lines connecting the respective first and last points of the sets. Then the algorithm is recursively applied to those two set-line pairs. This method is also described in [8] and [6].

3 Progressive line detecting

The transformation from the polar into the Cartesian coordinate system can be avoided using an algorithm that does not change the type of the coordinate system.

3.1 The Hough Transformation

This method was proposed in 1962 by Paul V.C. Hough[4] to automatically detect the complex tracks of subatomical particles in photographs obtained from a bubble chamber. The basic idea is to transform a single point into the parameter space of the set of infinite lines running through it. This parameter space is called *Hough space*.

In the basic form presented here, the Hough transformation detects straight lines only. But as pointed out in [7], the approach can be generalized to detect arbitrary picture elements. The Hough transform has been used for automatic mapping and robot localization in a number of projects, such as [2] (to detect edges in occupancy grids obtained from ultrasonic sensor output) and [5] (using a range-weighted variant for laser scanner based localization).

In the scanner-centered polar coordinate system of a laser scan, a line is given by the function

$$d(\alpha) = \frac{r}{\cos(\alpha - \phi)} \qquad (1)$$

where r is the length of the orthogonal from the coordinate system's origin to the line, and ϕ is its inclination (see figure 2). The parameter pair (r, ϕ) yields a single point in Hough space.

Examining a single point (d_1, α_1) from the laser scan, the parameter function for all lines passing through this point can be found by solving equation 1 for r. This yields

$$r(\phi) = d_1 \cos(\alpha_1 - \phi) \qquad (2)$$

for variable ϕ. Equation 2 is called the Hough Transformation of a single point. Thus, the Hough Transformation of a line is a point, and the one of a point is a cosine curve.

Setting up the line parameter equation for a second point (d_2, α_2), an angle ϕ_0 can be detected with

$$r(\phi_0) = d_1 \cos(\alpha_1 - \phi_0) = d_2 \cos(\alpha_2 - \phi_0) \qquad (3)$$

The pair $(r(\phi_0), \phi_0)$ is the parameter pair that specifies the straight line through (d_1, α_1) and (d_2, α_2). Its

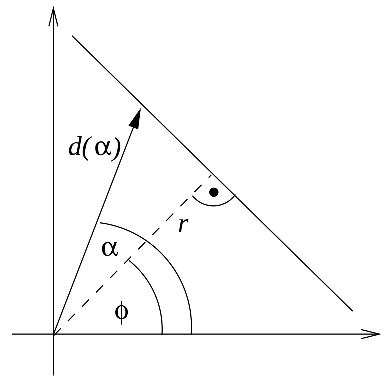

Figure 2: A line in polar coordinates with the specifying parameters

corresponding point in Hough space can be found by plotting the scaled cosine curves (obtained from equation 2 for (d_1, α_1) and (d_2, α_2)) and determining their intersection. Note that the second intersection point $(r(\phi_0 + \pi), \phi_0 + \pi)$ can be ignored because it yields the same line equation. See figures 3 to 5 for an illustration.

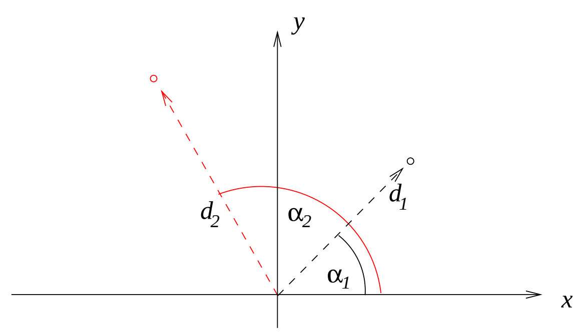

Figure 3: The points $p_1 = (d_1, \alpha_1)$ and $p_2 = (d_2, \alpha_2)$

Using this method, the equation for the line connecting an arbitrary number of collinear points can be found by transforming all the points into Hough space and determining the positive intersecting point of the resulting cosine curves. If the intersection search is limited to detect only the intersecting points of more than a certain number of curves, the Hough transform becomes very tolerant with respect to noise.

Unfortunately, in implementation, the Hough transform is not only robust and noise-tolerant but also fairly expensive in terms of memory and CPU time. The reason for the memory disadvantage arises from the fact that the Hough space has to be discretized in order to facilitate intersection search. The reason for CPU time consumption lies in the need to re-compute the cosine curve from equation 2 for every single point.

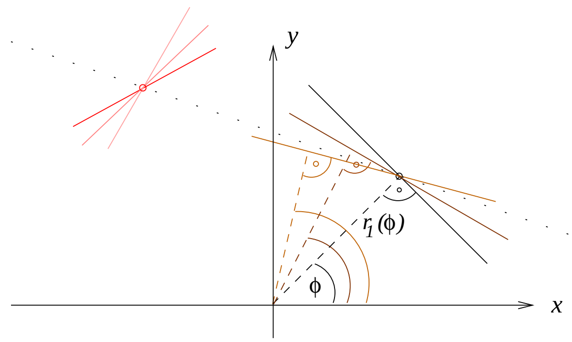

Figure 4: The parameters of the line sets through p_1 and p_2

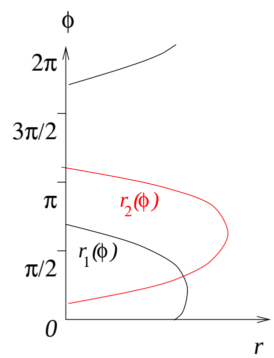

Figure 5: The representation of the parameter functions $r_1(\phi)$ and $r_2(\phi)$ in Hough space

3.2 The Log-Hough Transformation

The Log-Hough transformation as proposed by Weiman [11] is a modification of the Hough method. It greatly reduces both the amount of memory and the complexity of calculations required. This is achieved by performing all calculations not in a regular polar coordinate system but its logarithmized variant. Additionally, the used system is special in terms of being *isotropical*, i.e. the scale is equal to both axes. The demand for isotropy stems from the origins of the method: It was originally developed for processing data obtained from cameras with exponentially arranged CCD cells (see [10]), where such a coordinate system facilitates picture zoom and rotation simply by index shift.

The scale of the ϕ axis is given by the angular resolution of the scanner. Therefore the transformation is chosen such that the scale of the r axis is rendered appropriately. This is done by dividing the distance d by the inner measurement limit r_0, yielding equation 4 for the transformation into the log-polar coordinate system and equation 5 for the backwards transformation.

Note: For the purpose of the experiments conducted, the natural logarithm is used, so the exponentiation in the backwards transformation is written as e^x. In practice, however, any base can be used.

$$d_{log} = \log(\frac{d - r_0}{r_0}) \quad (4)$$

$$d = r_0 e^{d_{log}} + r_0 \quad (5)$$

The application of this transformation changes the straight-line parameter equation 2 to

$$\begin{aligned} r_{log}(\phi) &= \log(\frac{r(\phi)}{r_0}) \\ &= \log(d) - \log(r_0) + \log(\cos(\phi - \alpha)) \quad (6) \end{aligned}$$

From this equation, it can easily be seen that the point values d and α now only act as shifting parameters for the curve; the general shape stays the same, namely that of a logarithmized cosine curve. Therefore the implementation only needs to compute and discretize the curve exactly once. This prefabricated curve is deposited in a look-up table which is then plotted into Log-Hough space for each scan point, shifted by the point's parameters.

After transforming an entire scan, the intersection points are found in Log-Hough space. For each intersection, the angular parameter ϕ of the resulting line equation can be derived directly from the intersection's ϕ coordinate. The distance parameter r is obtained by transforming the r_{log} coordinate using equation 5.

In implementation (i.e. in a discretized grid with k cells in radial direction), the logarithmic distance coordinate is expressed as $d_{log} = n\delta$ where δ is the angular resolution of the scanner and n is the cell number that contains the detected intersection.

It remains to be noted that the linear discretization of Log-Hough space results in an exponentially proliferated discretization of real space. Therefore, accuracy decreases from the center ring (distance r_0 from the scanner) outward. However, since the area of greatest interest is about 2m in front of the robot, this is not a great disadvantage, especially when compared to the rate of data reduction that results from the logarithmized as opposed to the standard Hough transform. For an example, see section 5.3.

3.3 Termination

After all infinite lines existent in the scan have been extracted, their respective start and end points are found by again iterating over the entire scan and testing the distance of each point to each line against a threshold.

3.4 Uncertainty computation

To determine the uncertainty of the edge segments' properties, first the covariance matrix for the point

(r, ϕ) has to be determined. Unfortunately, this is not trivial. Linear regression in the polar coordinate system is unsatisfactory due to the non-linear relationship of the parameters, and the transformation of all generating points of a segment into the Cartesian system would remove all the benefits gained by the Log-Hough transform.

Instead, a simple approximation is used: The Log-Hough algorithm can be viewed as finding clusters in $(\epsilon_d, \epsilon_\alpha)$-space with

$$\epsilon_d = d - \frac{r}{\cos(\phi - \alpha)} \quad \text{and} \tag{7}$$

$$\epsilon_\alpha = \alpha - \phi + \arccos(\frac{r}{d}). \tag{8}$$

Now the variances of r and ϕ can be set equal to the variances of d and α as determined by

$$\sigma_r^2 = \frac{1}{n}\sum_{i=1}^{n}(d_i - \frac{r}{\cos(\alpha_i - \phi)} - \mu_d)^2 \tag{9}$$

$$\sigma_\phi^2 = \frac{1}{n}\sum_{i=1}^{n}(\alpha_i - \arccos(\frac{r}{d_i}) + \phi - \mu_\alpha)^2 \tag{10}$$

where $\mu_d = \frac{1}{n}\sum_{i=1}^{n} d_i$ and $\mu_\alpha = \frac{1}{n}\sum_{i=1}^{n}\alpha_i$. Then the covariance matrix of the point (r, ϕ) can be set to

$$\Sigma_{(r,\phi)} = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \tag{11}$$

This is a very rough estimate that has been chosen purely for performance reasons, but it performs well in practice.

A different approach would be to calculate the propagation of the single points' covariances to the covariances of each of the generated parameter points. Then, instead of discrete Log-Hough lines with height 1, lines made up of the approximated Gaussian covariance curves would be plotted. This would make it possible to obtain the uncertainties of the extracted lines directly from Log-Hough space. We are currently investigating this approach to find an efficient way to implement it.

After the variances of (r, ϕ) have been found, the point and the line's normal vector are converted into the Cartesian system and their covariance matrices are found. This information is used to parameterize the edge segment with start and end points, mid point and normal vector.

4 Mapping

Mapping is done in a Cartesian coordinate system with $(0, 0)$ at the point where the robot was located when the mapping process was started. Matching of already-modeled edges with those from a laser scan is done using the Mahalanobis distance[8], taking the edge features' covariances into account.

After an edge from the world model has been successfully matched with the corresponding edge from a new laser scan, the modeled edge is refined using a discrete-time Kalman filter[9]. Another Kalman filter is used to reduce the uncertainty in the robot's own dead-reckoning-based position estimation.

The resulting mapping algorithm produces a reliable model of a static environment. Methods for reaction to dynamic environments have not been integrated yet, but are subject to investigation.

5 Experimental results

This algorithm was tested with two different types of laser scanners. After a short description of the scanner hardware, an example is presented.

5.1 The scanner

Figure 6 shows the used scanner models. On the left hand side, the Sick PLS can be seen. This scanner performs a time-of-flight measurement with an accuracy of 5.1cm at a distance of up to 2m and 13.1cm beyond that.

Figure 6: The scanners from Sick and TRC

The other scanner shown is the HelpMate LightRanger. This unit is fitted with a laser emitting/detecting device from Acuity Research [1], which emits a square wave whose pulse width is modulated by the time of flight of the laser beam deflected from a rotating mirror. This square wave is sampled by an evaluating unit, transforming the pulse width into range readings.

Accuracy evaluations have shown the LightRanger's range output to have an accuracy of about 6.7cm regardless of distance to the measured object.

5.2 Edge extraction using the Log-Hough Transform

In this section, a short demonstration will be given on the results obtained using the Log-Hough Transformation. On the left side of figure 7, the state of the

Log-Hough grid after transforming all the points from figure 1 is depicted.

Figure 7: The Log-Hough space and its line maxima for figure 1

The right side of figure 7 shows the maxima of the discretized Log-Hough space's lines; this plot could be described as a "side view" of Log-Hough space. First, the peak-searching algorithm looks for peaks in this one-dimensional grid to determine the angular position of the respective peak. Then it finds the radial position by using the same one-dimensional maximum search on the corresponding line of Log-Hough space.

Finally, after the peaks have been determined and the corresponding lines have been terminated, the edge segments are transformed into a Cartesian coordinate system. This yields the edge segments shown in figure 8.

5.3 Data reduction

To see the factor of data reduction that arises from using the Log-Hough algorithm as opposed to the standard Hough transform, an example shall be given using some standard parameters.

Memory reduction. It is assumed that the area of interest stretches from $r_0 = 50$cm to $r_1 = 1,000$cm or 10m in radial direction and encompasses 360° of angular field of view. The angular resolution of the laser scanner be $\delta = 1/2°$. Discretization of this area with a minimum accuracy of 1cm in standard Hough

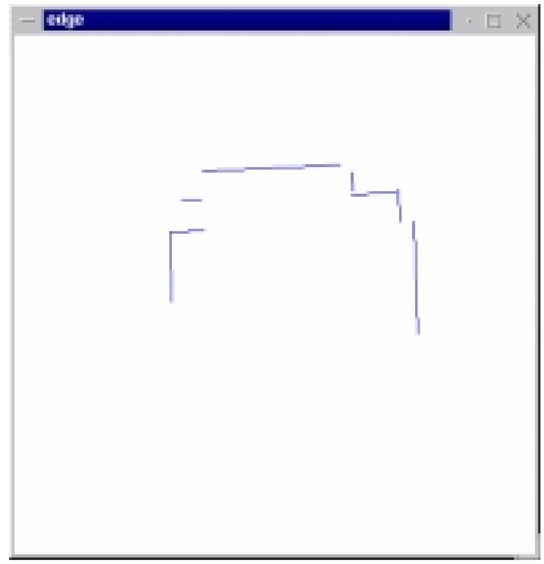

Figure 8: Edge segments extracted from figure 1 using the Log-Hough transformation.

space requires a grid of

$$(1,000 - 50) * \frac{360}{\frac{1}{2}} = 684,000 \qquad (12)$$

memory cells. If a cell size of a single byte is used, this grid is prohibitively large for many embedded systems.

Examining the grid required for a Log-Hough transformation with the same parameters, it can be found from the discretized form of equation 5 that

$$\frac{r_1 - r_0}{r_0} = \frac{950}{50} = e^{k \cdot \frac{1}{2}°}. \qquad (13)$$

Therefore the number of rings is $k = \lceil \log(19)/0.5° \rceil = 338$ if the natural logarithm is used. Multiplied with the number of angular wedges, this yields a grid size of

$$338 * \frac{360}{\frac{1}{2}} = 243,360 \qquad (14)$$

memory cells, or a data reduction of almost the factor 3.

The resolution of this Log-Hough grid computes to

$$\Delta d_0 = e^{2\delta} - e^{\delta} = 0.0088m \qquad (15)$$

or about 9mm in the innermost and

$$\Delta d_1 = e^{338\delta} - e^{337\delta} = 0.1659m \qquad (16)$$

or about 16.6cm in the outermost ring. So the minimum resolution has been retained close to the sensor, and in 10m distance the resolution is about 1.3 times the variance of the PLS at that range, or 2.5 times the variance of the LightRanger.

Calculation reduction. Compared to the commonly-used algorithms to extract edges, the Log-Hough transform saves CPU power as well.

The effort of the line tracking algorithm is $O(n)$. It is slowed down, however, by the transformation of the laser scanner data into the Cartesian coordinate system. The least squares method is very expensive also.

The iterative endpoint fit has the effort $O(n \log n)$, and it also requires a coordinate transformation.

The Log-Hough method as presented in this paper is an algorithm with effort $O(n)$, but without any complex coordinate transformations, making it the fastest algorithm available for the extraction of edges from laser scanner data.

6 Conclusion and future work

This new approach speeds up edge extraction significantly, especially since it is apt for realization in hardware due to the lack of complex transformations required. It is therefore ideal for automated mapping and position correction. Currently only 2D-mapping is realized.

For the future, it is planned to integrate some kind of three-dimensional sensor into the system, extending the Log-Hough method to fully encompass three-dimensional space. The sensor used for this will probably be a Dornier EBK laser scanner.

Another field of interest with this algorithm will be position correction. This can be achieved quite easily by detecting principal peaks in Log-Hough space and tracking their offset over several scans.

ACKNOWLEDGMENTS

Many thanks are due to HelpMate Robotics Inc. in Danbury, CT, USA, where much of the work for this paper was conducted, and to the company's founder Joseph Engelberger, who offered us that possibility.

We are especially indebted to Carl Weiman, Ph.D., who invented the Log-Hough transformation with the aid of NASA and NSF funding [3][11], and who supervised the work at HelpMate Robotics. Without his help, this work would not have been possible.

The final parts of the work were performed at the Institute for Process Control and Robotics (IPR), Prof. Dr.-Ing. U. Rembold, Prof. Dr.-Ing. H. Wörn, Prof. Dr.-Ing. R. Dillmann, Faculty for Computer Science, University of Karlsruhe.

References

[1] Inc. Acuity Research. AccuRange 3000-lv and AccuRange 3000-lir User's Manual. Technical report, Publication of Acuity Research, Inc., Menlo Park, CA, USA, Aug 1993.

[2] J. L. Crowley B. Schiele. A comparison of position estimation techniques using occupancy grids. In *Robotics and Autonomous Systems*, volume 12, pages 163–172, 1994.

[3] S.J. King C.F.R. Weiman. Log Hough Transform Feature Detector, Phase I SBIR Final Report, NASA Contract No. NAS 9-18489. Technical report, HelpMate Robotics Inc., 1994.

[4] Paul V.C. Hough. Method and Means for Recognizing Complex Patterns, U.S. Patent No. 3,069,654. Technical report, HelpMate Robotics Inc., Dez 1962.

[5] Å. Wernersson: J. Forsberg, U. Larsson. Mobile Robot Navigation Using the Range-Weighted Hough Transform. In *IEEE Robotics & Automation Magazine*, Mar 1995.

[6] Weckesser P. and Dillmann R. Sensor-fusion of intensity- and laserrange-images. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 501–508. Washington, USA, 1996.

[7] P.E. Hart R.O. Duda. Use of the Hough Transformation to Detect Lines and Curves in Pictures. In *Communications of the ACM*, volume 15, pages 199–218, 1972.

[8] P.E. Hart R.O. Duda. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[9] George M. Siouris. *An Engineering Approach to Optimal Control and Estimation Theory*. J.Wiley & Sons, 1983.

[10] Carl F.R. Weiman. Exponential Sensor Array Geometry and Simulation. In *Proc. of the SPIE Conf. on Digital and Optical Shape Representation and Pattern Recognition*, Apr 1988.

[11] C.F.R. Weiman. Application of Log-Hough Transform to Lidar Navigation, Phase I SBIR Final Report, NSF Award No. III-9361990. Technical report, HelpMate Robotics Inc., 1994.