IEEE ISIC'02

Some Sensing and Perception Techniques for an Omnidirectional Ground Vehicle With a Laser Scanner [1]

Zhen Song, YangQuan Chen, Lili Ma and You Chung Chung

Center for Self-Organizing and Intelligent Systems (CSOIS),
Dept. of Electrical and Computer Engineering, 4160 Old Main Hill,
Utah State University, Logan, UT 84322-4160, USA.

Abstract This paper presents some techniques for sensing and perception for an omnidirectional ground autonomous vehicle equipped with a laser scanner. In an assumed structured environment, the sensing data processing methods for both 1D and 2D laser scanner are discussed. Raw data are segmented to lines, circles, ellipse, planes and corners by task depended segmentation algorithms. Each subset of data is then fit by a known template shape as listed above. With these medium level information, the vehicle can infer its relative position with respect to the known landmarks and in turn help to determine its absolute position on the map.

Key Words: Hough transform; line fitting; corner fitting; arc/circle fitting; ellipse fitting; algebraic fitting; weighted circle fitting; Omni-directional vehicle (ODV).

1 Introduction

1.1 Motivation

For any intelligent mobile robot, the perception of its environment via suitable sensing capacities plays a key role [1, 2]. The environment perception depends largely on the properties of environment itself. Basically, the environment can be assumed to be either static (deterministic) such as office corridor or dynamic (changing) such as a parking lot with vehicles in and out dynamically. In terms of deterministicism of the environment, there are two types: deterministic environment and uncertain environment. With combination, we have four types of environments: static deterministic, dynamic deterministic, static uncertain and dynamic uncertain. For static and dynamic deterministic environments, the sensing and perception task is easier with a help of a map and the known motion patterns of the objects in the environment. For uncertain environment, it seems no *off-the-shelf* solution exists for general sensing and perception tasks for mobile robots [1]. This is because, sensing and perception solutions are usually *platform-specific* for mobile robot with given or existing sensing capability [1, 2]. However, there are still some common and possibly low-level solutions applicable for all platforms. This paper attempts to present some commonly used techniques for sensing and perception for mobile robot equipped with a 1D or 2D laser scanner in static uncertain environment. Our test platform is an omni-directional ground autonomous vehicle developed in the CSOIS at Utah State University (USU).

Our approach for static uncertain environment perception is based on the following steps (1) laser scan raw data, (2) segmentation, (3) fitting from the object library (or template objects), (4) object arbitration or decision. For step-1, pre-processing is performed to reject some obviously abnormal points or outliers. In step-2, the range data array from a laser scan is to be segmented to form some subsets of point clouds containing essential information for the object fitting. We assume that a library of objects such as line, corner/rectangle, arc/circle/ellipse etc. are available from the prior knowledge about the static uncertain environment. Then in step-3, we fit the through the library since we do not know in advance which object the given point cloud represents. The arbitration or decision on which object in step-4 is done by a rule based comparison of the fitting scores. We believe that this procedure is natural and can be regarded as a common piece of utility in sensing perception using a laser scanner. In each step of the above proposed procedure, there are many technical as well as theoretical challenges. In the present paper, we shall more focused on step-3.

The major contribution of this paper is the collection of some commonly used object fitting algorithms well tested in our experiments with our C++ and Matlab codes publically available [1].

1.2 The USU ODIS/T4 Platform

The USU ODIS robot is a small, man-portable mobile robotic system that can be used for autonomous or semi-autonomous inspection under vehicles in a parking area [3]. Customers for such a system include military police (MP) and other law enforcement and security entities. The robot features (a) three "smart

[1] Jan. 2002. *For submission to IEEE International Symposium on Intelligent Control (ISIC'02)*, Oct. 2002, Vancouver, Canada, as an invited session paper. Invited Session organized by Dr Jason Gu. This work is supported in part by U.S. Army Automotive and Armaments Command (TACOM) Intelligent Mobility Program (agreement no. DAAE07-95-3-0023. Corresponding author: Dr YangQuan Chen. E-mail: yqchen@ieee.org; Tel. 01-435-7970148; Fax: 01-435-7972003. URL: http://www.crosswinds.net/~yqchen.

[1] In the final version of this paper, we will give an URL for downloading the object fitting codes developed in this paper.

wheels" in which both the speed and direction of the wheel can be independently controlled, (b) a vehicle electronic capability that includes multiple processors, and (c) a sensor array with a laser, sonar and IR sensors, and a video camera. ODIS employs a novel parameterized command language for intelligent behavior generation. A key feature of the ODIS control system is the use of an object recognition system that fits models to sensor data. These models are then used as input parameters to the motion and behavior control commands. Fig. 1 shows the mechanical layout of the ODIS robot. The robot is 9.8 cm tall and weighs approximately 20 kgs. Key ODIS subsystems include its mechanical, vehicle electronics (vetronics) and sensor systems. For more detailed description, see [4] [5].

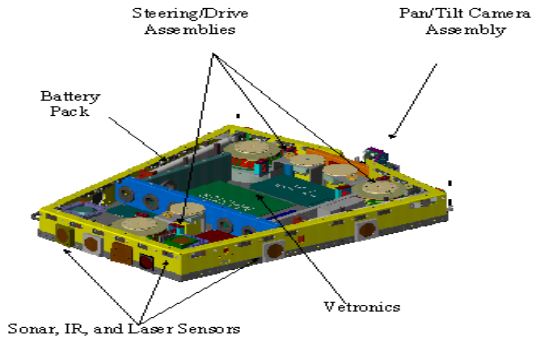

Fig. 1: The mechanical and vetronics layout of ODIS

Fig. 2 shows the behavior control architecture that has been developed. Starting from the "inside out," the control architecture contains two inner motion-control loops. The inner most loop is the wheel-level control, which acts to drive each smart wheel to its desired steering and drive speed set-points. The wheel-level controller uses simple PID control algorithms. Around the inner loop is the path-tracking controller. This loop derives the set points need by the wheel-level control in order to force the vehicle to follow a desired path in space, where a path is defined as an arc in inertial space (with a prescribed velocity along the arc) and an associated vehicle yaw motion. The path-tracking controller uses a newly-developed spatial tracking control algorithm that is described in more detail in [5].

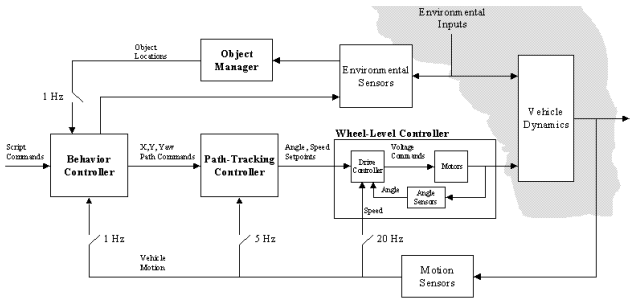

Fig. 2: The behavior control system architecture of ODIS

Fig. 3 shows the 1D laser scan data for a typical set of objects in front of the ODIS equipped with a 1D laser. In our project, objects in parking lots can be decomposed into combinations of arcs/circles and lines/rectangles.

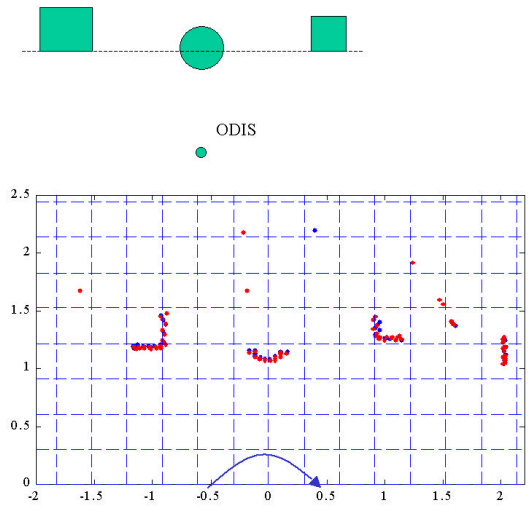

Fig. 3: 1D Laser Data (Top: Object Setting. Bottom: Laser Data)

Currently, the cost of a 2D laser scanner is greatly, which enables increasingly many robots be equipped with 2D laser scanner. Again, "*off-the-shelf*" algorithms for object identification or for environment features extraction have not been available although some very platform-specific schemes are proposed or tried, e.g., the research work reported in [6, 7, 8, 9, 10, 11, 12]. Fig. 4 shows the coordination system of our 2D SICK laser [2]. For computational convenience, it is different from the body frame coordination system (BFCS) and inertial coordination system (ICS). The height of 2D laser above ground, denoted by h, is constant. From the data of 2D laser scanner, we can get a set of raw data points (x', y') together with the corresponding pitch angle α from the tilting servo system. To convert the raw data into 3D coordination system (x, y, z), we have $\beta = \text{atan}(y'/x')$ and $d = \sqrt{x'^2 + y'^2}$. So, we get $(x, y, z) = (d\cos(\beta)\sin(\alpha), d\sin(\beta), h - d\cos(\alpha)\cos(\beta))$. In Sec. 3, we will introduce some object fitting algorithms such as plane fitting (Sec. 3.4), 3D corner detection and fitting (Sec. 3.5), and etc. for 2D laser scanner.

The rest of the paper is organized as follows. Sec. 2 presents a revised Hough transform method for an efficient segmentation of the laser scan raw data set. Also in this section, lamp pole simulation data are presented to give an example of task depended segmentation. A set of fitting algorithms are described in Sec. 3 which includes both 1D and 2D laser data processing such as line or plane fitting, circle fitting, 3D corner fitting etc. Finally, Sec. 4 concludes this paper with some remarks on our on-going research efforts.

[2] `http://www.sickoptic.com/kommerce_server/`

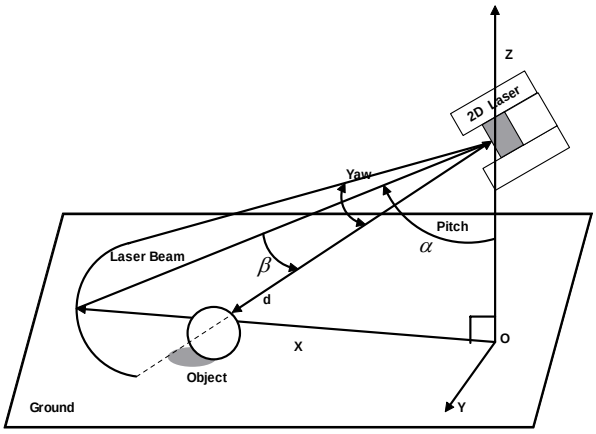

Fig. 4: 2D Laser Coordinate System

2 Segmentation

2.1 Line Detection for Segmentation

Line detection is the first step to segment 1D laser data. After the line detection, we can judge if a point is on the line by checking the distance between the point and the line. In our project, it is reasonable to segment objects in parking lots to circles and lines. So segmentation is simplified as line detection and circle fit.

There are many paper published on line detection. Some took the frequency-domain approach [13] and others took time-domain approach, e.g. Hough Transform [14], or color pattern analyze [15]. In our project, the input is sequential laser data, instead of image. In the following discussion, we will focus on Hough Transform.

Hough Transform (HT) is a popular method for the extraction of geometric primitives [14, 16]. At the beginning, it is only an approach for line detection. Later, many variant are developed, thus this algorithm can detect circle [16], ellipse [17], or more complex binary patterns [18]. The survey on Hough Transform [14] gives a good big-picture on the current research progress.

Generally speaking, HT is robust to sensor noise at the expense of slow computation. The compensational cost of the traditional HT is $O(n^3)$ where n is the number of data points. Though some efforts were made to speed up the HT algorithm [19], those fast strategies were developed generally for image data processing applications only.

However, in our case, 1D laser data set contains only *sparse* binary information in the region of interest. Furthermore, only 1D array is requred for laser data instead of 2D matrix for image data. More importantly, the laser data set is sequential or ordered. Taking these properties of laser data set, HT can be modified to reduce the computational time.

2.2 Sparse Hough Transform for 1D Line Detection

In order to make the most use of those valuable features of laser scanner data set, here we propose a Sparse Hough Transform (SPHT) that fit better to the laser data processing problem. The Standard Hough Transform (SHT) represents lines crossing a point by

$$\rho = x\cos\theta + y\sin\theta,$$

where ρ and θ are the length of the normal and the angle of the normal with respect to the positive x-axis. x and y are the coordinates of a point of interest in the image. ρ and θ are quantified as the dimensions of an accumulator matrix. The value of each cell of the matrix represents how many times there is a line, denoted by (ρ, θ), passed a point in the image. The idea of Sparse Hough Transform is to provide an equivalent but faster and less memory-demanding approach. Basically, SPHT does not use a 2D array to record the votes. Instead, it uses a 1D array to do the same vote-counting job. Since in sparse case, most lines passed one point are not the fitted lines, we can compute only those lines that passed at least two points in the image. Transfer all the lines corresponding to one point in the image plant, as SHT does, would take extra computer memory and reduce the speed.

Based on the above considerations, we present an implementation of SPHT by the following pseudo-code list:
(`Dat` is the input array; `Dat(n).x` is the x value of the n-th point. `Lines` is the output array that stores (ρ, θ) of the fitted lines. `NumLen`=0, which is the number of fitted lines.)

```
for i=1 to n-1
   for j=i+1 to n
      α = atan((Dat(i).y−Dat(j).y)/(Dat(i).x−Dat(j).x))
      ρ0 = (Dat(i).x cos(α -π/2) + Dat(j).y sin(α − pi/2)
      θ= α-sign(ρ0) π/2
      ρ = |ρ0|
      Cellθ=floor(θ/Δθ)*Δθ
      Cellρ=floor(ρ/Δρ)*Δρ
      If line n, where n ∈ [1,NumLen] is close enough to
(Cellρ,Cellθ) increase the vote of Lines(n) by one, otherwise
NumLen=NumLen+1 and add a line (Cellρ, Cellθ) to Lines with
vote equates to one.
   end for
end for
```

Fig. 5 shows an example of the proposed SPHT. The SPHT transforms or fits the raw set data to 2 lines. Actually, the raw data could be interpreted as 2 lines, or 2 lines plus a small transit arc, or even just one line. The exact segmentation depends on the resolution, or the threshold. For example, with a more precise laser sensor, we can know that the data in Fig. 5 cannot be a single line. However, if the data is from a low-precision sonar sensor, the target object might be just a single line. With a given sensor in real project, we could choose some reasonable thresholds by referring to the specifications of the sensors. Further discussions on the trade-off between sensor accuracy and the object fit accuracy can be found in [20] and [21].

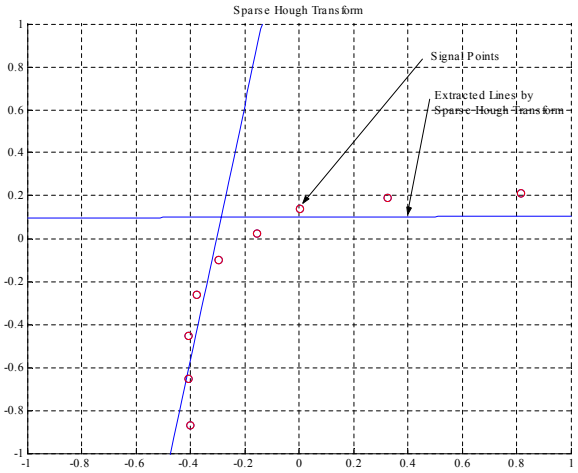

Fig. 5: An Example of Sparse Hough Transform

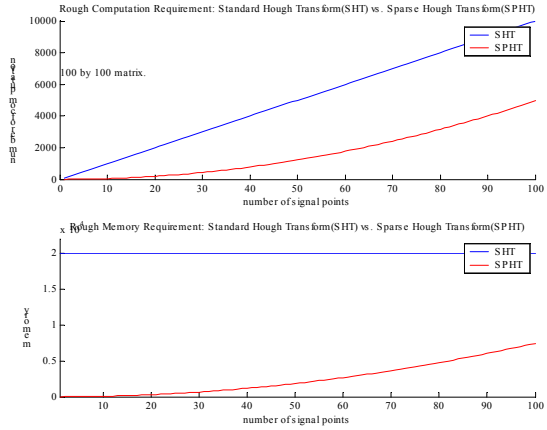

Fig. 6: Speed Comparison of SPHT and SHT

2.3 Analysis on Speed and Memory Size

In general, suppose the angle resolution is given by $\Delta\theta$ and normal resolution $\Delta\rho$. For SHT, $M \times N$ cells are required for the accumulation space, where $M = \pi/\Delta\theta$ and $N = \rho_{max}/\Delta\rho$. In other words, the memory size is $n \times M \times N$, where n is the size of each cell. If there are K valid points in the input data, the computation requirement is $O(K \times M)$.

For SPHT, if there are only K valid points in the input, clearly $K \ll M$ and $K \ll N$. The maximum number of fitted lines could be $K \times (K-1)/2$. The computation requirement is then $O(K \times (K-1)/2)$.

Figure 6 shows the computation cost of SPHT versus SHT when the image matrix/array has 100 by 100 points. In practice, in a laser scan, the number of data point is in the order of 10, i.e., the array contains typically only tens of valid data points. From Fig. 6, clearly, SPHT has a better performance over SHT.

2.4 Task-Dependent Segmentation

To process 2D laser data, segmentation is more complicated than in the case of 1D laser data. The idea of task-dependent segmentation is to simplify the situation by make a better use of the known information. Currently, we use lamp pole and side walk as landmarks to calibrate the odometry system of the mobile robot.

Lamp pole is an important landmark in parking lot. Fig. 7 is the simulated data of a lamp pole from a 2D laser scanner mounted on T4, the mother robot or vehicle for ODIS. Here we assumes the lamp pole is normal to the ground and ground is flat. The task is to segment the raw data so that the position of the center of the lamp pole (cylinder) can be computed by some fitting algorithms. Some of these computations were done by Matlab functions. Fig. 8 is the image that the lamp pole projected to the ground. Here we actually separate the ground from the lamp pole. Fig. 9 is the result of edge detection of Fig. 8. Fig. 11 is the 3D object when we re-project the edge in Fig. 9 back to 3D space. At last, we set a threshold to eliminate those points too close to each other, then we get Fig. 11.

After projecting these points to the ground we get a circle with the same center of the 3D lamp pole. In other words, when we feed the result of the segmentation to our fitting algorithms, we can get the position of the feature points. Then the relative position of the landmark is available which can be used for mobile robot odometry calibration, or localization.

Another important landmark is the convex corner of the side walk. Since the height of side walk in a specific parking lot is constant, we can set a threshold to segment the ground and upper layer. The segmentation is straightforward and we will not discuss it here due to space limitation.

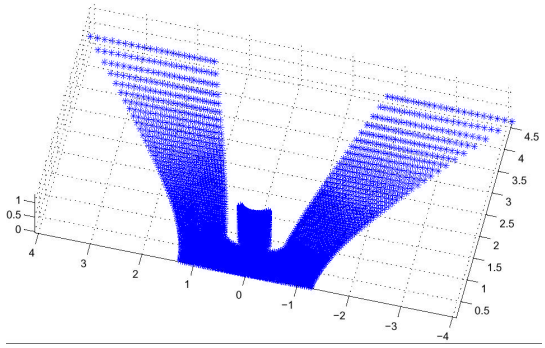

Fig. 7: Lamp Pole Simulation Data

3 Template Fitting Algorithms

With properly segmented data sets from a laser scan, it is not certain which template in the template library is to be used for the object fitting, without human involvement. A viable way is to try fit all templates in the library and then pick the "best" one via an arbitrator based on some rules from the possible prior knowledge

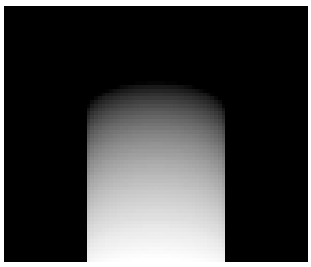

Fig. 8: Projected Lamp Pole Image

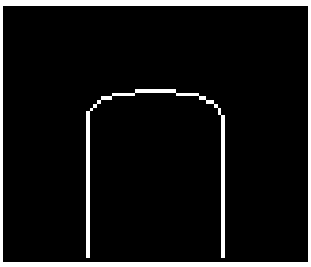

Fig. 9: The Edge Detection on Lamp Pole Image

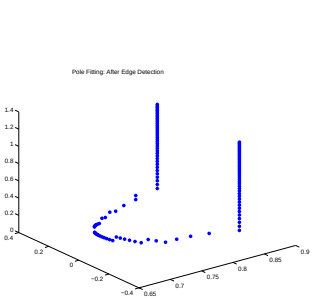

Fig. 10: Reverse The Edge of Lamp Pole Image

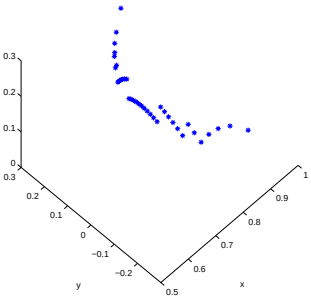

Fig. 11: Lamp Pole Data After Segmentation

about the objects in the static and uncertain environment.

For computational efficiency, we focus ourself on the algebraic fitting which does not require any iteration. For geometrical fitting, is may make more sense but the iterations may take a long time and some times, convergence cannot be guaranteed. It is of course, when possible, beneficial to use geometrical fitting result to cross-validate that from the algebraic fitting.

Most of our codes for fitting are tested in Matlab and then converted to C++ [3] which are ready for running in our Linux box.

3.1 Algebraic Circle Fit

When the radius of a circle is unknown, a simple algebraic circle fitting method can be applied to find the best fit circle in the least squares (LS) sense.

Given a set of points with coordination $\{(x_i, y_i), i = 1, 2, \cdots, n\}$, find the best parameters $a_1, \cdots, a_4$ in the circle equation

$$a_1(x^2 + y^2) + a_2 x + a_3 y + a_4 = 0,$$

such that the algebraic error is to be minimized. The center of the circle is (x_c, y_c) with $x_c = -\frac{a_2}{2a_1}$, $y_c = -\frac{a_3}{2a_1}$.

[3] We used Dr. Robert Davies' free C++ library "`NewMat`" downloadable from `http://webnz.com/robert/cpp_lib.htm`.

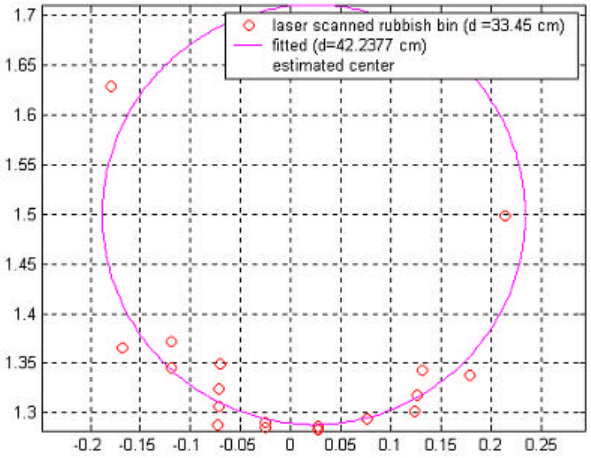

Fig. 12: Circle Fit

Let $\mathbf{a} = [a_1, a_2, a_3, a_4]'$. Construct the matrix $\mathbf{D}$ as

$$\mathbf{D} = \begin{bmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^2 + y_n^2 & x_n & y_n & 1 \end{bmatrix}.$$

Then, the LS solution of equation

$$\mathbf{Da} = 0$$

is simply given by the SVD (singular value decomposition). See Fig 12 for a typical fit to an experimental 1D laser scan data set.

3.2 Circle Fit with Known Radius

Some of the round objects in the static uncertain environment may have known radii. So, although simpler, it is practically desirable to perform circle fit with known radius. Here we use geometrical error for the fit performance index.

Given a set of points with coordination $\{(x_i, y_i), i = 1, 2, \cdots, n\}$, find the best parameters x_c and y_c, the center coordinate of the circle

$$(x - x_c)^2 + (y - y_c)^2 - R^2$$

where R is the known radius, such that the geometric error $\varepsilon = \sum_{i=1}^{n} (\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - R^2)$ is to be minimized.

Via the gradient of ε with respect to x_c and y_c, an iterative scheme for x_c and y_c can be formulated and implemented. Since R is known, the convergence process is quite reliable and fast.

3.3 Ellipse Fit

The general form of an ellipse is described by

$$a_1 x^2 + a_2 xy + a_3 y^2 + a_4 x + a_5 y + a_6 = 0.$$

The ellipse fit problem is very similar to the algebraic circle fit method presented in Sec. 3.1.

Here the matrix sent for SVD is given by

$$\mathbf{D} = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix}$$

and $\mathbf{a} = [a_1, \cdots, a_6]'$

3.4 Plane Fit

Different from the above fitting method mainly for 1D laser scan data set, the methods of this subsection and the next subsection are designed for 2D laser scanner data processing. Actually, the acquired data from a 2D laser scanner is a 3D array, as mentioned in Sec. 2. The goal of the plane fit problem is to find the optimal coefficients a_1, a_2, a_3, a_4, such that

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = 0$$

has a solution in LS sense. Again, here we use the SVD technique for the plane fitting.

3.5 3D Corner Fit/Detection

Corner detection is normally studied in the context of 2D image processing. Corner is one of the basic image features and the corner detection has been extensively studied [22, 23]. For 3D corner detection, we can project the 3D corner to a 2D surface and detect the corner in 2D. The advantage of this method is simple. However, some times it may not be able to detect some corners in certain 3D configurations. Here we propose two effective 3D corner fit algorithms.

The first method fits an intersection point of 3 planes as the corner. So there is not limitation on whether the corner is convex or concave. Moreover, the corner is not necessarily to be symmetric. The disadvantage of this algorithm is that it can not fit corner without a prior plane segmentation step.

This corner fit problem is to find the best corner $\{x^*, y^*, z^*\}$, such that

$$\begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \\ \vdots & \vdots & \vdots \\ A_n & B_n & C_n \end{bmatrix} \times \begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_n \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

or $M \times P = -D$, where, $e_1^2 + e_2^2 + e_3^2$ is to be minimized the minimum. Based on the projection theorem, the solution is

$$P = (M^T M)^{-1} M^T (-D)$$

$$[\, x^* \quad y^* \quad z^* \,] = P^T.$$

The second algorithm can fit corners directly from 3D point set. This approach reduces the requirement on segmentation. We can separate points to small subsets, and then fit them by the following corner fit function. The limitation of the algorithm is that the corner function is only suitable for convex and symmetric corners. The proposed corner function is described by

$$z = e^{k_1(x-x_0)^2 + k_2(y-y_0)^2 + k_3}$$

where k_1, k_2, k_3, x_0, y_0 are parameters to be fit. In LS sense, we can re-formulate the above fitting problem into

$$\begin{bmatrix} x_1^2 & x_1 & y_1^2 & y_1 & 1 \\ x_2^2 & x_2 & y_2^2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_n & y_n^2 & y_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} \log(z_1) \\ \log(z_2) \\ \vdots \\ \log(z_n) \end{bmatrix}.$$

And starting from here, it will be the same as in the previous method.

In the final version of this paper, we will include the section on "Arbitrator" or "Object Decision" as well as the related experimental results.

4 Concluding Remarks

This paper presents some techniques for sensing and perception for an omnidirectional ground autonomous vehicle equipped with a laser scanner. In an assumed structured environment (static but uncertain), the sensing data processing methods for both 1D and 2D laser scanner are discussed. Raw data are segmented to lines, circles, ellipse, planes and corners by task depended segmentation algorithms. Each subset of data is then fit by a known template shape as listed above.

Our immediate effort is to make use of these medium level information in our vehicle to infer its relative position with respect to the known landmarks and in turn help to determine its absolute position on the map, a procedure known as mobile robot localization. Other future efforts include (1) the motion estimation of dynamic obstacle(s) by assuming that the environment is dynamic uncertain; (2) the fusion with sonar, laser scanner and image sensing information for local map building and (3) relative navigation without absolution position information (or inertial/world coordinates) via sensor fusion and spatial filtering.

Acknowledgment The authors would like to acknowledge the fruitful discussions with CSOIS members in Vetronics Group and Intelligent Behavior Group. In

particular, the authors are grateful to Professor Kevin L. Moore, Director of CSOIS, for his support of this work.

References

[1] M. D. Adams, *Sensor modelling, design and data processing for Autonomous navigation*, vol. 13 of *World Scientific in Robotics and Intelligent Systems*, World Scientific, Singapore, 1999.

[2] I. R. Nourbakhsh, *Interleaving planning and execution for autonomous robots*, The Kluwer International Series in Engineering and Computer Science: Robotics: Vision, Manipulation and Sensors. Kluwer Academic Publishers, Boston, 1997.

[3] K. L. Moore and N. S. Flann, "A six-wheeled omnidirectional autonomous mobile robot," *IEEE Control Systems*, vol. 20, no. 6, pp. 53–66, 12 2000.

[4] K. Moore, N. Flann, Rich S., M. Frandsen, Y. Chung, J. Martin, M. Davidson, R. Maxfield, and C. Wood, "Implementation of an omni-directionsl robotic inspection system (ODIS)," in *Proceedings of SPIE Conference on Robotic and Semi-Robotic Ground Vehicle Technology*, Orlando, FL., May 2001, SPIE.

[5] M. Davidson and Bahl V., "The scalar ϵ-controller: A spatial path tracking approach for ODV, Ackerman, and differentially-steered autonomous wheeled mobile robots," in *Proceedings of IEEE Int. Conference on Robotics and Automation*, Seoul, Korea, 2001, IEEE.

[6] D. Dedieu, V. Cadenet, and P. Souères, "Mixed camera-laser based control for mobile robot navigation," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[7] H. Surmann, K. Lingemann, A. Nuchter, and J. Hertzberg, "A 3D laser range finder for autonomous mobile robots," in *Proceedings of the 32nd ISR2001 (International Symposium on Robotics), Seoul, Korea*, 2001, pp. 153–158.

[8] R. M. Taylor and P. J. Probert, "Range finding and feature extraction by segmentation of images for mobile robot navigation," in *Proceedings for the 1996 IEEE International Conference on Robotices and Automation Minneapolis, Minnesota*, Apr. 1996.

[9] J. Vandorpe, H. V. Brussel, and H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder," in *Proceedings for the 1996 IEEE International Conference on Robotices and Automation Minneapolis, Minnesota*, Apr. 1996.

[10] V. Cadenat, P. Souéres, and M. Courdesses, "Two milti-sensor-based control strateties for driving a robot amidst obstacles," in *Proceedings for the 39th IEEE Conference on Decision and Control Sydney, Australia*, Dec. 2000.

[11] J. Laurent, M. Talbot, and M. Coucet, "Road surface inspection using laser scanners adapted for the high precision 3D measurements of large flat surfaces," in *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling IEEE*, 1997.

[12] J. Borenstein, H. R. Everett, and L. Feng, "Where am I, sensors and methods for mobile robot positioning," `http://www-personal.engin.umich.edu/~johannb/position.htm`, 1996.

[13] K. Abed-Meraim and Y. Hua, "Multi-line fitting and straight edge detection using polynomial phase signals," IEEE-IP, Dec. 1997.

[14] J. Illingworth and J. Kittler, "A survey of the Hough Transform," *CVGIP*, vol. 44, pp. 87–116, 1998.

[15] J. Puzicha and S. Belongie, "Model-based halftoning for color image segmentation," in *Interational Conference on Pattern Recognition*, Barcelona, Spain, 2000, pp. 629–32.

[16] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *Image and Vision Compution*, vol. 17, pp. 795–803, 1999.

[17] N. Guil and E. L.Zapata, "Lower order circle and ellipse Hough Transform," *J. Pattern Recognition*, vol. 30, no. 10, pp. 1729–1744, Oct. 1997.

[18] N. Guil and E. L. Zapata, "A new invariant scheme for the generalized Hough transform," in *IASTED Int'l. Conf. on Signal and Image Processing*, Orlando, Fl,, Nov. 1996, pp. 88–91.

[19] J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic Hough Transform," in *Proc. British Machine Vision Conference BMVC98*, Sep. 1998.

[20] H. Goto, "The efficient sampling interval of the scanning parameter in the Hough transform," *Systems and Computers in Japan*, vol. 29, no. 11, pp. 697–705, Apr. 1998.

[21] H. Goto and H. Aso, "Designing efficient Hough transform by noise-level shaping," *IEICE Trans. Inf. and Syst.*, vol. E83-D, no. 2, Feb. 2000.

[22] M. A. Ruzon and C. Tomasi, "Corner detection in textured color images," in *IEEE Seventh International Conference on Computer Vision*, Sep. 1999, pp. 1039–1045.

[23] S. M. Smith and J. M. Brady, "SUSAN - a new approach to low-level image processing,," in *International Journal of Computer Vision*, 23(1): 1997, pp. 45–78.