International Journal of Innovation and Scientific Research
ISSN 2351-8014 Vol. 27 No. 1 Oct. 2016, pp. 1-8

http://www.ijisr.issr-journals.org/

A Novel Line Extraction Algorithm using 2D LiDAR for Ego-motion Estimation

Lei Ye

College of Mechatronic Engineering and Automation,
National University of Defence Technology,
ChangSha, Hunan, China

ABSTRACT: The paper proposes a convex hull-based algorithm for rapid line extraction from 2D LiDAR datas. It uses an algorithm to calculate feature points in LiDAR data frames. Geometric features contained in these feature points provide information for subsequent matching. Compared with traditional LiDAR matching algorithms, the algorithm is greatly improved in terms of iterations and matching precision. This algorithm finally was used to solve the ego-motion estimation of an indoor robot.

KEYWORDS: scan matching; feature extraction; mobile robot; ego-motion estimation

1 INTRODUCTION

1.1 MOTIVATION

Accurate estimation of motion parameters is important for the safety and task execution of intelligent robots [2]. Traditionally, robot ego-motion parameters are estimated in terms of speed, angle and position by means of an inertial sensor carried onboard. Inertial information is calculated from multiple integrals of parameters such as acceleration and angular velocity. These are easily affected by sensor errors, which are unbounded. Perceptual sensors, gathering information via systems such as LiDAR, are now widely used in intelligent robots to determine their position and motion.

1.2 PROBLEM DESCRIPTION

Computing the motion parameters of robots using LiDARs requires the transformation parameter, T , to be estimated by matching two frames of LiDAR data. Generally, two frames of data are represented by the model $\{\mathrm{m}_i\}_{i=1}^{N_m}$ and current data $\{S_i\}_{i=1}^{N_s}$. Resolution of ego-motion parameters is a process of transforming $\{S_i\}_{i=1}^{N_s}$ by using T to achieve the best alignment with $\{\mathrm{m}_i\}_{i=1}^{N_m}$.

If it is supposed that the robot is moving on a two-dimensional plane, Euclidean transformation of a three-dimensional parameter, $a=[\theta,t_x,t_y]$, is used to represent the motion parameter of the robot. The problem can be described as:

$$T(a;x)=T(\theta,t_x,t_y;X)=\begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}X+\begin{pmatrix}t_x\\ t_y\end{pmatrix} \quad (1)$$

The effect of alignment can be evaluated by an error function, which can be defined as:

$$\varepsilon^2(|X|)=||X||^2 \quad (2)$$

The foundation of alignment is the correspondence between $\{m_i\}_{i=1}^{N_m}$ and $\{S_i\}_{i=1}^{N_s}$, which can be represented by $\phi(i)$, and so we can derive the error function：

$$E(a,\phi)=\sum_{i=1}^{N_d} w_i \varepsilon^2(|m_{\phi(i)}-T(a;s_i)|) \quad (3)$$

The parameter w_i represents the coincidence relation. It is set to 1 if there is a coincidence relation between points in the model, otherwise it is set to 0. We can see that $\phi(i)$ is also a parameter in the error equation, and generally we resolve the problem with iterative optimization. Specifically, we should resolve $\phi(i)$ on the basis of $\{m_i\}_{i=1}^{N_m}$ and $\{S_i\}_{i=1}^{N_s}$:

$$E(a)=\sum_{i=1}^{N_d} w_i \min_j \varepsilon^2(|m_j-T(a;s_i)|) \quad (4)$$

Then we calculate the ego-motion parameter $a=[\theta,t_x,t_y]$ on the basis of obtaining $\phi(i)$:

$$\hat{a}=\arg\min_a \sum_{i=1}^{N_d} w_i \min_j \varepsilon^2(|m_j-T(a;s_i)|) \quad (5)$$

1.3 RELATED RESEARCH

There has been extensive research on the estimation of ego-motion parameters derived from LiDAR systems. The Iterated Closest Point method (ICP) [1] put forward by Besl [3] and Zhang [4] is the most well-known. The ICP algorithm makes use of a nearest-neighbor search to solve the best coincidence relation for the current motion of the robot ($a=[\theta,t_x,t_y]$) in the process of computing $\phi(i)$. The method does not make use of the characteristics of the data itself, so it is easy to fall into a local minimum solution when the estimation of initial parameters is inaccurate. Another relatively well-known method is the NDT algorithm proposed by [5], which first meshes the data, then represents $\{m_i\}_{i=1}^{N_m}$ and $\{S_i\}_{i=1}^{N_s}$ with a multi-dimensional Gaussian distribution in each grid, and finally estimates ego-motion parameters using a method similar to ICP. Compared with ICP, the method performs feature extraction from raw data and has clear superiority in terms of operating speed. However, the size of the grid has a large influence on experimental effects and operating speed. Accuracy declines if the selected grid is too large, and the operating speed declines if the selected grid is too small. It is thought that ICP is a special case where the grid is very small in NDT. Due to the shortcomings in these raw data processing methods, much research on feature matching has subsequently been conducted.

1.4 MAIN CONTRIBUTIONS AND OUTLINE OF THIS PAPER

In this paper, a new feature extraction algorithm for 2D LiDAR sensing systems is proposed. Feature points within a single frame of LiDAR scanning data can be extracted online through linearization of LiDAR data. Because these feature points represent environmental information within a certain error range, they can be used to estimate the ego-motion parameters of robots.

This paper is structured as follows: Section Two reviews common linearization methods and compares their principles and effects. Section Three outlines the methods proposed in this paper and the steps required for their implementation. Firstly, a convex hull-based algorithm for rapid line extraction is described. Then, feature selection and feature association, based on linear portion segmentation results, are investigated. Finally, a nonlinear optimistic algorithm is derived to solve for ego-motion parameters. Section Four discusses the operating effects of linearization methods and the results of the solutions for ego-motion parameters.

2 PRIOR WORK

Many studies have been conducted on the image processing, data mining and 2D LiDAR sensing aspects of line extraction methods, which mainly involve the algorithms below:

The Split-and-Merge algorithm can be considered the oldest and most frequently used line segmentation method [6,7,8]. The algorithm has been used in image processing, feature detection and positioning, based on 2D LiDAR data. If the last point of a line is selected, the algorithm is called Iterative-End-Point-Fit, and a linear portion can be obtained under a finite number of iterations through continuous segmentation.

The concept of an incremental algorithm [9], [10], [11] is very simple. Basically, new points are constantly added to a current straight-line model to see whether it continues to meet its line constraints. Straight-line models are generally calculated with the least square method, and criteria to identify whether it meets the line constraints are established from the maximum and total error distances between new points and the straight line. During actual implementation, to improve implementation efficiency, n points are added to the current straight line model, and will continue to be added as long as it meets the constraints. The n points added should be eliminated if the straight-line model constraints are exceeded, and should be added one by one.

The RANSANC [13] algorithm is not only used in line detection, but also widely used for the detection of various models in the field of computer image processing. The basic concept is to select several points of concentrated objects to construct the model, then detect which points belong to the model and which do not. Points that belong to the model are called interior points, and those do not are called exterior points. Interior points are used to calculate the model. If there are sufficient exterior points, one should continue to carry out these steps. The RANSANC approach was initially used in the construction of environmental maps and for robot positioning based on the processing of LiDAR data [12].

The Hough transform algorithm was initially used for line detection in images and was extended to detect various shapes including circles, ellipses and even non-parametric contours. The biggest difficulty of the Hough variation method lies in selecting a suitable parameter scale. Smaller parameter resolutions result in increased parameter space, which not only increases the calculated amount, but also generates false targets due to decentralized voting. The Hough variation method is mainly used in the processing of 2D LiDAR data, and map building [14], [15].

3 Approach

3.1 Convex hull-based algorithm for rapid line extraction

This paper proposes a new algorithm for line extraction, which calculates the convex hull (CH) of data from a segment of discrete points. It surrounds the convex hull with two parallel linear portions. The two parallel lines obtained are linear portions to be solved. See the schematic diagram in Fig. 1.

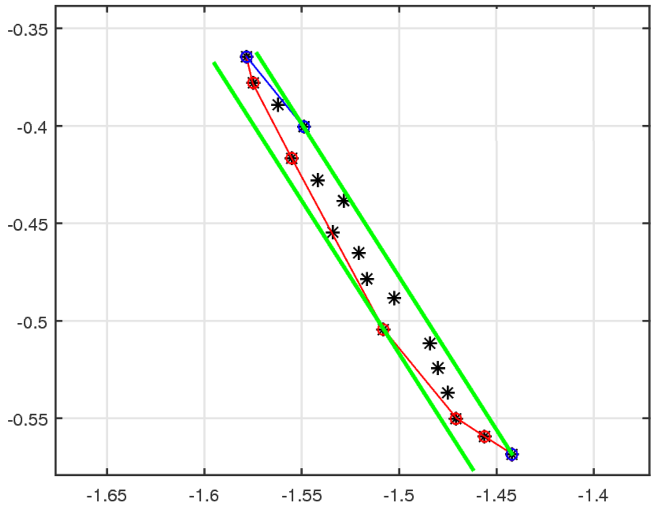


Fig. 1. schematic diagram of a convex hull

The black asterisks in Fig. 1 represent the original sampling points. The red line represents the outer contour of the data point set. The blue line (partly shaded) represents the inner contour, and the green lines represent the two extracted parallel line segments. The inner and outer contours represent contours far from, and near, the LiDAR. It can be seen from the schematic diagram that all original sampling points are distributed between the two parallel lines which tightly clamp them to achieve an efficient representation of discrete sequence points.

The classic convex hull algorithm is mainly used for convex hull solutions, and is not used much anymore. The new convex hull algorithm uses the extensions of a point-adding method and a divide-and-conquer algorithm. The divide-and-conquer algorithm, also known as QUICK-HULL (QH), is widely used in the calculation of two-dimensional and three-dimensional convex hulls. Many scholars hope to use parallel computing methods to improve the calculation speeds of convex hulls. The consolidation method in [16] can be used for parallel calculation of 2D and 3D convex hulls. In terms of data structure, the sub-results can be recorded with decision trees [17]. The parallel algorithm mentioned in [18] has reduced the complexity to $O(\log n)$, and extended its application from 2D to 3D.

LiDAR data is ordered in the direction of θ; therefore, the sorting process of the classical algorithm above is not needed. The contour furthest from the LiDAR is defined as the outer contour of the convex hull: H_{out}, and the one closest to the LiDAR is defined as the inner contour: H_{in}. Every time a new scanning point appears, it can be identified as belonging to $CH[P]$ or not. The convex hull algorithm for computing the complexity, $O(n)$, can be obtained by using the "turning right" characteristics of classical convex hull algorithms which do not need parallelization, and each scanning point needs to be calculated once. At the same time, in the process of calculating the convex hull, the linear portion to be solved can be calculated online. The algorithm is as follows:

Algorithm：Line detection algorithm based CH

Input：Point set $\{P_i\}_{i=1}^{N}$,$Dis_{Threshold}$

Output: $\{DP_i\}$

Initialization: $DP_1 = P_1$, $cnt = 1$

While $i \le N$

  Update convex hull, after add P_i to current point set；

  Calculate the width of convex hull: $Width_{CH} = Dis(H_{in}, H_{out})$

  If $Width_{CH} > Dis_{\text{Threshold}}$

    $cnt = cnt + 1$

    $DP_{cnt} = P_i$

End

The implementation process is demonstrated in Fig. 2.

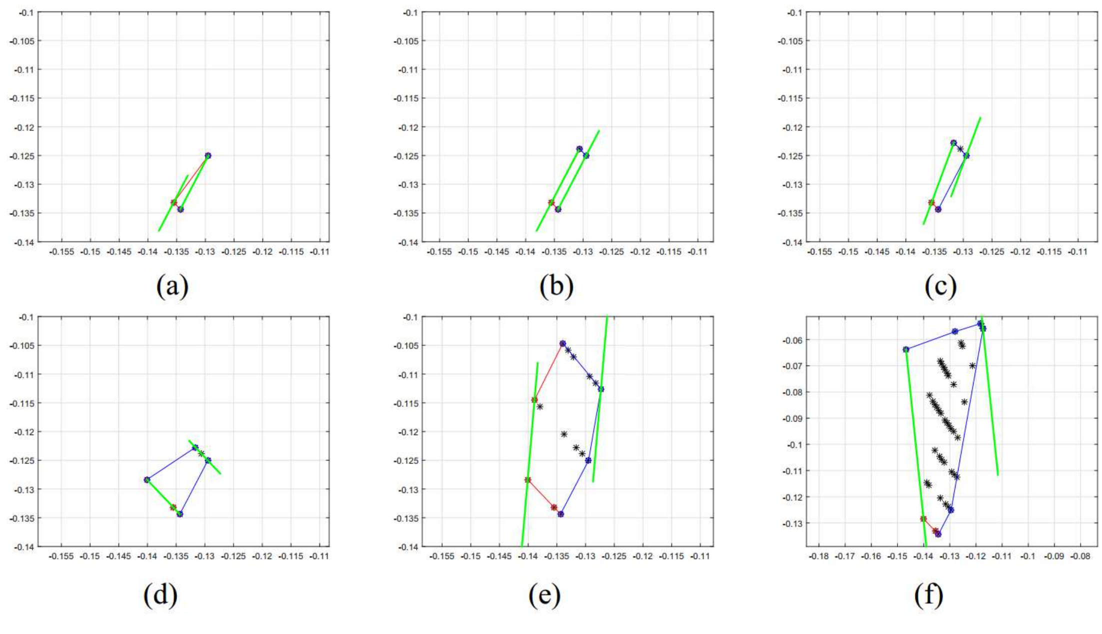


Fig. 2. process of a convex hull

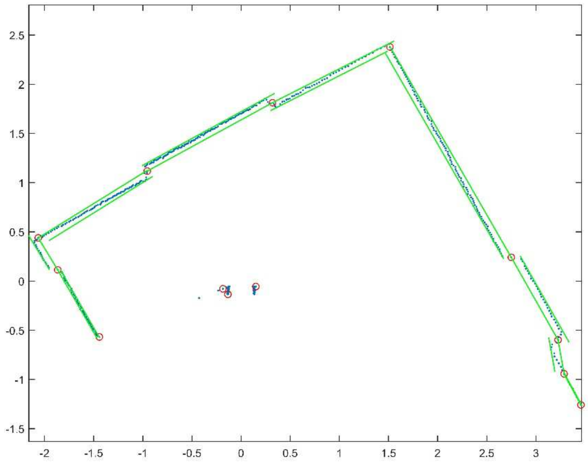


Fig. 3. result of our algorithm

Fig. 3 shows the results of our algorithm. The red points represent the spatial information of the environment.

3.2 FEATURE SELECTION

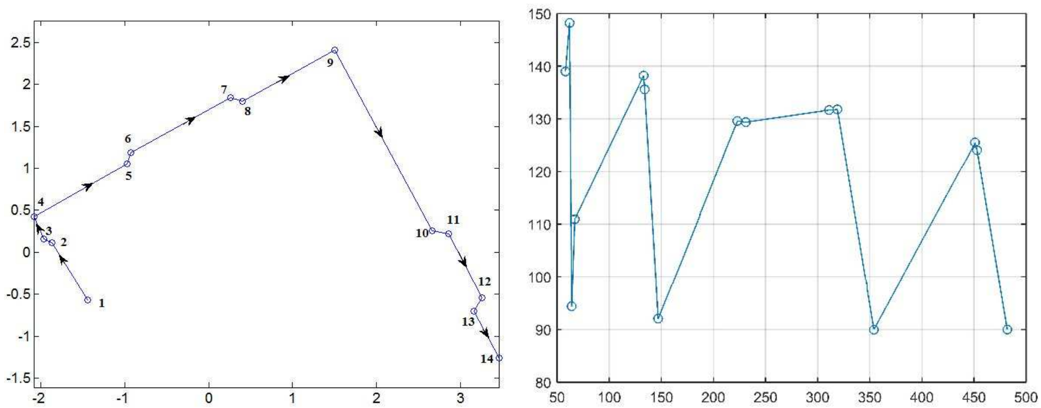
The convex hull-based 2D LiDAR data rapid line extraction algorithm proposed above can use two types of image features, i.e., linear portions and dominant points (Fig. 3). As the dominant point is the end point of a linear portion, the dominant point information represents the linear portion information completely. Accordingly, this paper will use dominant points to estimate robot ego-motion parameters.

After extracting the dominant point from the sequence point cloud, the dominant point is seen to be ordered and directional. That is, the dominant point is extracted clockwise and stored according to the extraction sequence, as shown in Fig. 4 (a). In order to take advantage of the special characteristics of the dominant point, the most intuitive feature is the angle between the two consecutive lines. Three adjacent dominant points constitute the angle θ.

The angle θ formed by the three dominant points can be easily obtained by using the law of cosines:

$$\theta = \mathrm{acos}\left(\frac{\overrightarrow{l_{ij}} \cdot \overrightarrow{l_{jk}}}{\left\|\overrightarrow{l_{ij}}\right\| \left\|\overrightarrow{l_{jk}}\right\|}\right) \tag{6}$$

Obviously the characteristic angle has rotation invariance, translation invariance and scale invariance. Fig. 4(b) shows the characteristic angle of the corresponding dominant points.



(a) (b)

Fig. 4. characteristic angle

Besides, the discrete curvature of the feature points can be selected as a feature, and can be calculated according to the method defined in [16]. That is, take the circle with P_i , P_j and P_k on it to approximate the osculating circle with P_k on it on the curve segment. As the radius of the circle represents the radius of curvature, the computing formula is as follows:

Assume that:

$$\begin{cases} s_1 = length(l_{ij}) \\ s_2 = length(l_{jk}) \\ s_3 = length(l_{ik}) \end{cases} \tag{7}$$

then,

$$\gamma = \frac{s_1 s_2 s_3}{\sqrt{(s_1+s_2+s_3)(-s_1+s_2+s_3)(s_1-s_2+s_3)(s_1+s_2-s_3)}} \tag{8}$$

Fig. 5 is the discrete curvature of the corresponding dominant points in Fig. 4.

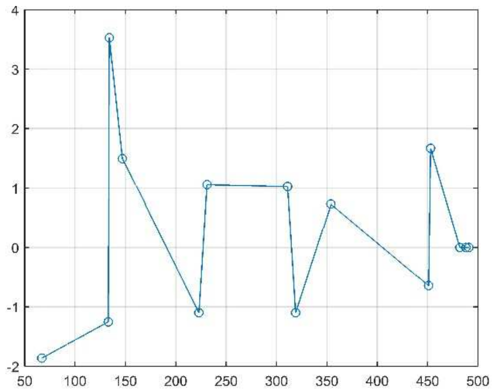


Fig. 5. discrete curvature

3.3 FEATURE ASSOCIATION

Assume that the feature points calculated in each frame data are static. Then, the relative motion of feature points between frames can be used to calculate the ego-motion parameters of a robot. So, it is very important to match the feature points from different frames. Feature data matching is usually based on the feature information contained in feature data, and the distances between feature points. Feature point matching is also known as data association, and has shown much promise for the navigation and positioning of robots. Common methods include threshold filtering, near neighbor data association, probability data association and multidimensional assignment data association. In this paper, multidimensional data iterative association [19] has been used.

The feature points proposed in the paper include the four-dimensional characteristic quantities of x , y , included angle θ , and discrete curvature γ . Together they constitute the characteristic quantity of a feature point: $[x, y, \theta, \gamma]'$

The weighted distance between two feature vectors can be defined as follows:

$$Dis_{ij} = \sqrt{w_1(x_i - x_j)^2 + w_2(y_i - y_j)^2 + w_3(\theta_i - \theta_j)^2 + w_4(\gamma_i - \gamma_j)^2} \tag{9}$$

Following the idea of the threshold method, if Dis_{ij} is less than a certain threshold, it is thought that a data association is established. However, due to the existence of sensor noise, the feature points between the two frames do not match perfectly, we use fractional correlation method for data association. Firstly, make use of data features which can be associated to solve the ego-motion parameters, and then transform current feature points to the coordinate system of a previous time for data association. After transformation, the feature points are much closer to the feature points to be matched, and are easily matched.

3.4 EGO-MOTION PARAMETER SOLUTION

After the feature association process, the correspondence function, ϕ_i , from Equation 5 has been solved. The solution to Equation 5 is actually the solution to the unconstrained optimization problem.

For non-linear unconstrained optimization problems [20], three methods are typically used, i.e. the quasi-Newton method, simplex search and non-linear least squares fit. This paper uses the quasi-Newton method. When using the quasi-Newton method to compute a partial derivative, the search direction is the decline direction of the function value. Through second-order Hesse matrices do not necessarily to be positive definite, the approximate Hesse matrices should be calculated, and Hesse matrices need not to be re-calculated in the iteration. Because the approximate Hesse matrices can be updated, the calculation is faster.

4 RESULTS

We constructed an experimental system with a Pioneer mobile robot and SICK 511 LiDAR. We then tested the method proposed in this paper in an indoor environment. The experimental results are shown below:

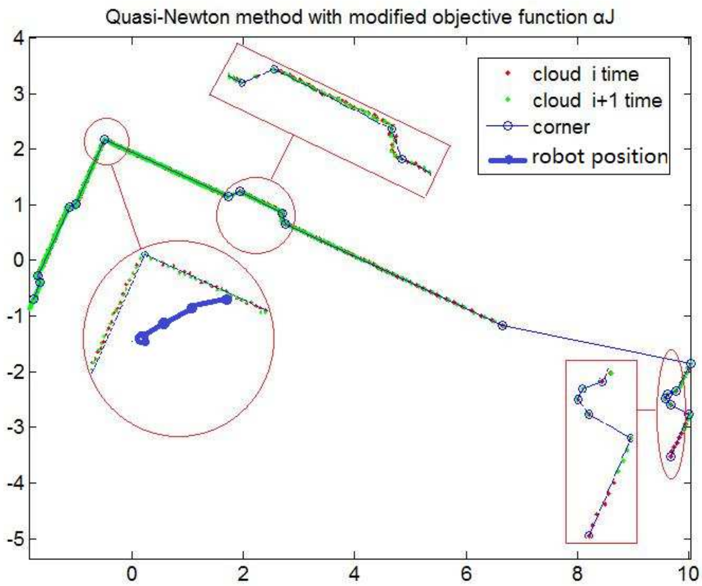


Fig. 6. matching result

It can be seen from Fig. 6 that that method our proposed method accurately calculates the robot's ego-motion. The accuracy is an improvement on the traditional ICP method.

5 CONCLUSIONS

The paper formulated a new line extraction algorithm that selects feature points from 2D LiDAR images. It solves the ego-motion parameters of an indoor mobile robot following a quasi-Newton method. This method calculates information from feature points and performs data association between frames. Experimental results show that the proposed method is superior to the commonly-used ICP algorithm, in both implementation efficiency and effect.

REFERENCES

[1] S Rusinkiewicz, Levoy M. “Efficient variants of the ICP algorithm”. *IEEE Computer Society*, 2001.

[2] M Dissanayake, P Newman, et al. “A solution to the simultaneous localisation and map building (SLAM) problem”. *IEEE Transactions on Robotics & Automation*, 17(3):229--241, 2001.

[3] P J Besl, N D Mckay. “Method for registration of 3-D shapes”. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(2):239-256, 1992.

[4] Z Zhang. “Iterative point matching for registration of free-form curves and surfaces”. *International Journal of Computer Vision*, 13(2):119-152, 1992.

[5] M Martin, L Achim, D Tom. “Scan Registration for Autonomous Mining Vehicles Using 3D-NDT”[J]. *Journal of Field Robotics*, 24(10):803-827, 2007.

[6] G A Borges, M J Aldon. “Line Extraction in 2D Range Images for Mobile Robotics”. *Journal of Intelligent & Robotic Systems*, 40(3):267-297, 2004.

[7] J Castellanos, J Tardos. “Laser-Based Segmentation and Localization for a Mobile Robot”. *Robotics & Manufacturing: Recent Trends in Research & Applications*. 2014:4.

[8] Einsele, Tobias. “Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder”. *International Conference on Intelligent Robots and Systems*. 1997:697-702 vol.2.

[9] A Siadat, S Klausmann. “An Optimized Segmentation Method For A 2d Laser-Scanner Applied To Mobile Robot Navigation”. 1999.

[10] D A Forsyth, J Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[11] R M Taylor, P J Probert. “Range finding and feature extraction by segmentation of images for mobile robot navigation”. *IEEE International Conference on Robotics and Automation*, 1996. Proceedings. 1996:95 - 100.

[12] J Vandorpe, H Van Brussel, H Xu. “Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder”. *IEEE International Conference on Robotics and Automation*, 1996. Proceedings. 1996:901-908 vol.1.

[13] C A C Mach. “Random Sample Consensus: a paradigm for model fitting with application to image analysis and automated cartography”. 1981.

[14] P Jensfelt, H I Christensen. “Laser Based Position Acquisition and Tracking in an Indoor Environment”. *Proc.int.symp.robotics & Automation*, 1998.

[15] S T Pfister, S I Roumeliotis, Burdick J W. “Weighted line fitting algorithms for mobile robot map building and efficient data representation”. 1:1304-1311 vol.1, 2003.

[16] M De Berg, M Van Kreveld, M Overmars, et al. Computational geometry: algorithms and applications. Library Philosophy & Practice, 2008, 2011(1):333-334.

[17] C B Barber, D P Dobkin, H Huhdanpaa. “The quickhull algorithm for convex hulls”. *Acm Transactions on Mathematical Software*, 22(4):469-483, 1996.

[18] F P Preparata, S J Hong. “Convex hulls of finite sets of points in two and three dimensions”. *Communications of the Acm*, 20(2):87-93, 1977.

[19] W S Wijesoma, L D L Perera, Adams M D. “Toward multidimensional assignment data association in robot localization and mapping”. *IEEE Transactions on Robotics*, 22(2):350-365, 2006.

[20] L E Scales. *Introduction to Non-Linear Optimization*. Maemillan, 1985.