A Line Segments Extraction Based Undirected Graph from 2D Laser Scans

Xinzhao Li [1], Yuehu Liu, Zhenning Niu, Zhichao Cui

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, No. 28 Xianning West Road, Xi'an, Shannxi, China

[1] mjperhaps@stu.xjtu.edu.cn

Abstract—**A novel algorithm to find line segments is proposed from the sequence of points taken by a laser scan, which consists of over segmentation, undirected graph generation and line segments extraction. Firstly, the self-adaptive IEPF method is used to over segment raw points into sub-groups. Secondly, the undirected graph is generated based on merge probability function of sub-groups. Thirdly, according to the main edges, the energy function of undirected graph partition and corresponding minimization method, line segments are extracted. The primary contributions of the work are the main edge, undirected graph and minimization method of energy function, which are more proper to noisy laser scan data. The experimental comparison with 6 state-of-the-art methods, using 2D laser scan data obtained by laser profile sensor and laser range finder, shows better performance and robustness of the new algorithm.**

I. INTRODUCTION

Nowadays, the application of robots has changed our daily life. Humans are being replaced by varies of robots to deal with the tasks which involve high risk, high duplication and high precision. In order to cover more complex tasks, robots are becoming more and more complicated. But it's needed to organize the raw data acquired by sensors. The most applied sensors in mobile robots are cameras, infrared, radar and laser. And the last two are more precise and faster than others.

Laser data processing, as the fundamental issue of robot artificial intelligence, has applied many pattern recognition procedures and concepts to organize the raw data adaptively. There are two techniques which have been used in data organization: point-cloud based(3D) method and geometric feature based(2D) method. Compared to the former, feature-based method requires much less memory and still provides rich and accurate information. In numerous geometric primitives, line segment is one of the most simple and common features. Laser sensor works in indoor or outdoor artificial environment which has structured characteristics. So the objects have planar external surfaces which naturally lead themselves to be represented by line segments in 2D-plane.

Many works have been done to extract line segments directly from the 2D laser data. Even so, many problems still exist, such as being sensitive to noise and threshold, poor robustness and too many pieces.

Considering the problems above in existing algorithms, we propose a line segments extraction algorithm for 2D laser scan data using undirected graph in this paper. The flowchart of our approach is shown in Fig.1. First we get the over segmentation of raw points data. Then the merge probability of sub-groups is defined and an undirected graph is generated from them. Afterwards we construct the energy function of undirected graph partition and propose the corresponding minimization method to get segmentation result of line segments. Our approach can be used in workpiece detection of robotic arm, SLAM of mobile robot and navigation of unmanned vehicle.

We make three novel contributions as follow:

(1)The traditional iterative end point fit algorithm (IEPF) is improved with adaptive threshold and outlier exclusion.

(2)To the best of our knowledge, none of other approaches make use of undirected graph, which in our opinion is a novel and advantageous solution to against the noise in laser data.

(3)The concept of main edge is proposed to combine practical issue into undirected graph partition. Also the energy function of undirected graph partition and corresponding minimization method are constructed, which can improve the performance and robustness of line segments extraction.

The rest of the paper is organized as follow: in section 2 the line segments extraction problem is formulated. Then we briefly review some related work in section 3. Section 4 to section 6 describes the whole process of our line segments extraction algorithm. The experiments and results are represented in section 6. Finally, section 7 is the conclusions.

II. PROBLEM STATEMENT

It's assumed that while robot moves, laser sensor collects a 2D laser scan data in fixed time interval. The scan data is a 2D slice of environment composed of ordered points. If the sensor is laser range finder, the data is represented by $R = \{r_i | r_i = (\rho_i \cos\varphi_i, \rho_i \sin\varphi_i)\}_{i=1}^{m}$. (ρ_i, φ_i) is polar coordinates of the point detected by i-th scan line of radar. If the sensor is laser profile sensor, the data is given by $R = \{r_i | r_i = (x_i, y_i)\}_{i=1}^{m}$. (x_i, y_i) is cartesian coordinates of i-th contour point. To more situations, we integrate the two forms into the later one. It is needed to emphasize that the order of points data should be maintained.

Formally, the extracted line are defined as a set of line segments such as $L = \{L_k | (\kappa_k, \beta_k, x_k^a, y_k^a, x_k^b, y_k^b)\}_{k=1}^{N}$, where κ_k and β_k are cartesian parameters of k-th line; the terminate point of the k-th line segment is (x_k^a, y_k^a) and (x_k^b, y_k^b); N is the number of extracted line segments.

So the line segments extraction problem can be stated as: **for a 2D laser scan data R, how to calculate the line segment parameters L to fit the points in R most.**

2015IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Oct. 19-21, 2015, Xiamen, China.

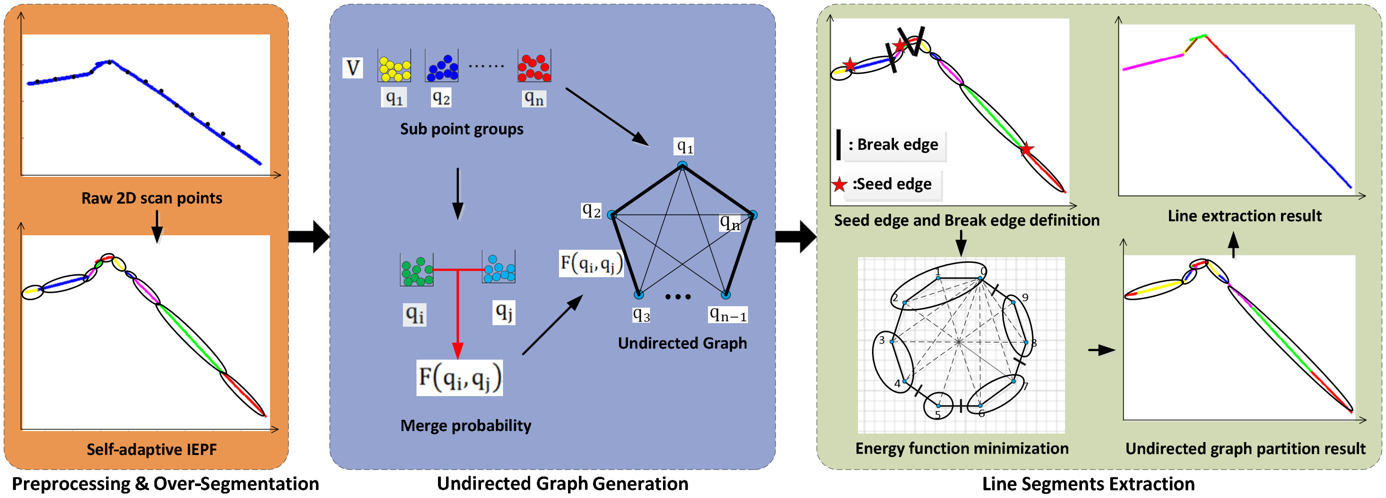


Fig. 1. The flowchart of proposed line segments extraction approach.

In order to accomplish line segments extraction algorithm we should pay attention to the three main problems as follow:

- How many lines are there?
- Which points belong to which line?
- Given the points that belong to a line, how to estimate the line segment parameters?

Among existing line segments extraction methods, they all have mentioned the concept of breakpoints intentionally or not. The breakpoints are associated to the discontinuities in the scanning process[1], which reflected to the computation are big distance, little probability or the rupture points. In our approach we also use the breakpoint in the undirected graph.

III. RELATED WORK

The problem of line segments extraction for 2D laser data has been solved in various different ways. Majority of methods can be simply classified into two categories: points based and figure based. And the points based methods can be divided to sequential or recursive algorithms.

Sequential algorithms make full use of the continuity of point sequence for line segments extraction. The essence is that a straight line can be composed only by continuous points. Successive edge following (SEF, also known as point distance based methods (PDBS)) [2] and line tracking (LT, also known as Incremental algorithm)[3,4] are well known in this method. These two methods both decide whether to add a new point into old sub group by judging whether the point meets the judgment criteria. What the difference is that SEF uses the distance of two adjacent points, however, LT use the distance between the point and fitting line. Nevertheless, these methods suffer from a high sensibility to the threshold and noise.

Recursive algorithms apply iteration to line segments extraction. As a representative, Split and Merge algorithm [5] divides the points into two sub groups by the breakpoint which is determined by the max distance between the point and the fitting line of points. It will become iterative end point fit (IEPF) [1][6][7] if the fitting line is created by simply connecting the first and last points. IEPF and its derivatives are commonly applied for line extraction in laser data after 1997[8]. Split and Merge Fuzzy (SMF) [1][7] is another recursive method which uses the fuzzy validation criterion based on fuzzy clustering prototype to determine whether to split points set. Unfortunately, recursive algorithms rely on the threshold as well as the sequential algorithms.

With the development of image processing, more and more image line extraction algorithm is proposed. So some scholars transform raw scan point data into image to make line detection. Hough transform(HT)[9] is a common technique for line-segment extraction for image. Forsyth [10] proposed the repeated HT method and Ogaz[11] proposed the advanced HT algorithm using minimun entropy. Other than HT, a famous algorithm called Fast Line Segment Detector (LSD)[12][13] has been applied to line extraction of laser data. Because the number of image pixel is far more than scan points, line detection in image is generally can't guarantee real-time performance. What's more, it will generate truncation errors when transforming from raw scan point data to image.

There are some other algorithms which don't belong to the former categories played a significant role in line extraction. RANSAC algorithm[14] is used with many types of features, so as line once we have the line model. EM algorithm[14] has also been used as a line extraction tool in robot. Furthermore, the mix of former algorithms has been used in line extraction. Such as SEF+LT[15] [16] and IEPF+HT[17].

IV. PREPROCESSING AND OVER-SEGMENTATION

As mentioned in section two, a frame of current 2D scan point data is obtained from laser sensor denoted by:

$$R = \{r_i | r_i = (x_i, y_i)\}_{i=1}^{m}$$

Firstly, several data preprocessings are done, which include zero point elimination and repeat point elimination. Because the limit of detecting range, laser will record zero when environment target is out of range. Furthermore, laser will record the same coordinate of different points due to the mechanical and signal problem. They will affect the consequence if we ignore these interference points.

Then the self-adaptive IEPF algorithm is used to coarsely divide the scan point data into sub groups(cluster). Because of the small value of threshold we used, this step is regarded as an over segmentation of R.

IEPF algorithm recursively split the set of points $R' = \{r_{m_i}, r_{m_{i+1}}, \dots, r_{m_e}\}$ into two subsets $R'' = \{r_{m_i}, \dots, r_{m_a}\}$ and $R''' = \{r_{m_a}, \dots, r_{m_e}\}$ if validation criteria $D_{m_a} < D_{MAX}$ isn't satisfied. r_{m_a} is the point whose distance D_{m_a} to the line formed by the extreme point of R' is maximum. The same procedure is repeated to R'' and R'''until all validation criteria are satisfied.

In the implementation, some improvements are made to get better and quicker result. The threshold is automatically adjusted according to the length of R' denoted by:

$$\mathrm{threshold}(R') = \mathrm{threshold}_0 + \lambda * \log_2 \frac{H(R)}{H(R')}$$

where λ is the step length of threshold and $H(R)$ is the length of point group R.

What's more, we will judge whether r_{m_a} is an outlier point. r_{m_a} will be excluded for another maximum if its adjacent points are not satisfied with the validation criteria.

V. Undirected Graph Generation

In this section an undirected graph is generated to translate the line extraction problem to graph partition problem.

After the over-segmentation, the sub groups V are got as:

$$V = \{q_k | q_k = \{r_{\sum_{i=1}^{k-1} n_k}, \dots, r_{\sum_{i=1}^{k} n_k}\}\}_{k=1}^{n}$$

where n_k is the number of points in k-th sub groups

Then the function of merge probability between two sub groups is defined as follow:

$$F(q_i, q_j) = \frac{f_r(q_i, q_j) + \alpha * f_k(q_i, q_j) + \beta * f_s(q_i, q_j) + \gamma * f_d(q_i, q_j)}{1 + \alpha + \beta + \gamma}$$

where q_i, q_j are two sub groups, α, β, γ are constant, and the other four items are defined as follow. Physical meaning of each item is show in Fig. 2.

$f_r(q_i, q_j)$ is the coarse IEPF merge item(Fig. 2(a)), which denotes the merge process by extending the threshold of IEPF. The formula of merge probability item is stated as follow:

$$f_r(q_i, q_j) = \begin{cases} 1 & D_{mid}(q_i) < D_{MAX} \cap D_{mid}(q_j) < D_{MAX} \\ T * \dfrac{\min(D_{mid}(q_i), D_{mid}(q_j))}{D_{MAX}} & D_{mid}(q_i) > D_{MAX} \cap D_{mid}(q_j) > D_{MAX} \\ T * \dfrac{\max(D_{mid}(q_i), D_{mid}(q_j))}{D_{MAX}} & \text{else} \end{cases}$$

where $D_{mid}(q_i)$ is the distance between the fitting line of IEPF and the far endpoint of sub group q_i. D_{MAX} is enlarged IEPF threshold. What's more, T is length constraint coefficient denoted by $T = \sqrt{\frac{D_{mid}(q_i)}{H(q_i)} * \frac{D_{mid}(q_j)}{H(q_j)}}$.

$f_k(q_i, q_j)$ is the kmeans cluster item(Fig. 2(b)). Here the average slope of sub group is calculated as clustering item. Because of the decrease of number and dimension after the over-segment, time-consuming of kmeans has been reduced greatly. The number of kmeans clusters is $\frac{n}{2}$ here.

$$f_k(q_i, q_j) = \begin{cases} 1 & q_i \in C_k \text{ and } q_j \in C_k \\ \dfrac{K(q_i) - O_{C_k}}{K(q_j) - O_{C_k}} & q_i \in C_k \text{ and } q_j \notin C_k \end{cases}$$

where C_k is k-th cluster of kmeans, $K(q_i)$ is the average slope of q_i and O_{C_i} is the average slope of cluster C_i.

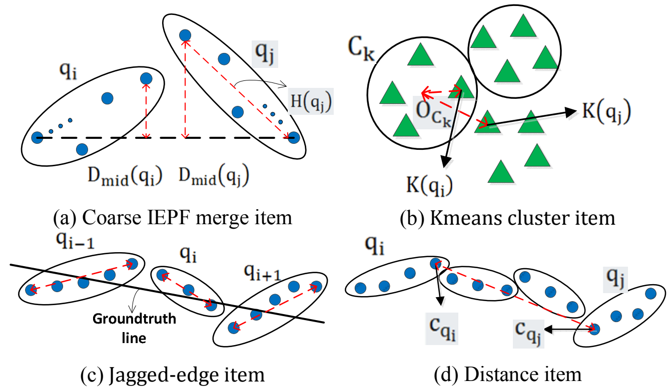


Fig.2. Physical meaning of the items in merge probability. The red dashed lines denote the variables used in items

$f_s(q_i, q_j)$ is the jagged-edge item(Fig. 2(c)), which indicates the jagged-edge shape of fitting line between adjacent point groups. If the adjacent point groups appear to the shape of jagged, we think the merge probability is bigger, which is defined as follow:

$$f_s(q_i, q_j) = \begin{cases} \sqrt{\dfrac{\left(\min_{k<n} H(q_k)\right)^2}{H(q_i) * H(q_j)}} & \begin{array}{l} j = i + 1 \text{ and } K(q_i) * K(q_j) < 0 \\ \text{and } K(q_i) * K(q_{i-1}) < 0 \end{array} \\ 0 & \text{else} \end{cases}$$

where q_{i+1} and q_{i-1} are the adjacent point groups of q_i, $\min_{k<n} H(q_k)$ is the shortest length of all the sub groups.

$f_d(q_i, q_j)$ is the distance item which measures the order of two point groups(Fig. 2(d)). It is defined by Gaussian kernel of points distance as follow:

$$f_d(q_i, q_j) = \exp\left(-\frac{1}{2\sigma} ||c_{q_i} - c_{q_j}||\right)$$

where $c_{q_i}, c_{q_j}$are the coordinates of nearest points in q_i,q_j.

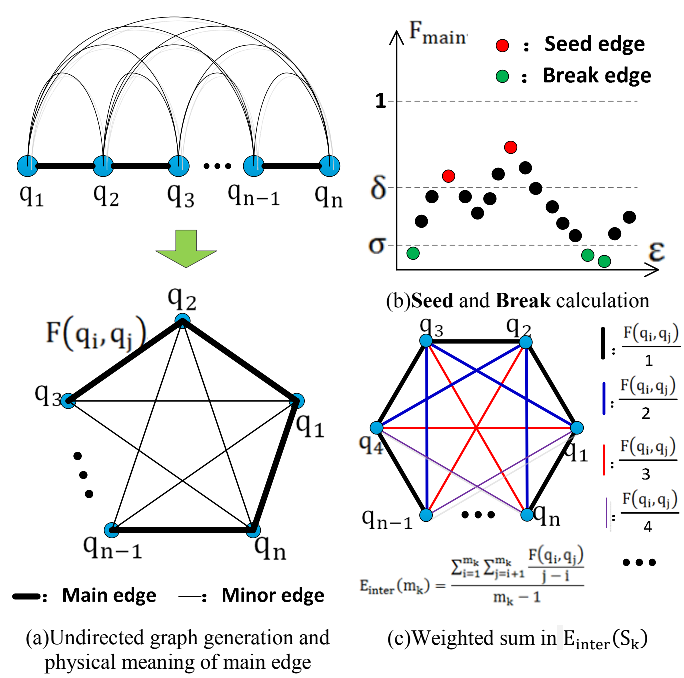


Fig. 3. Illustration of undirected graph

So an undirected graph $G=[v,\varepsilon]$ is generated as in Fig. 3(a). The nodes v stand for the sub point sets $V=\{q_1,\dots,q_n\}$, and the edges ε is the relationship of nodes defined as $F(q_i,q_j)$, the merge probability of two sub groups.

VI. Line Segments Extraction

After getting the undirected graph G of raw scan data, line segments extraction problem is changed into a math problem. The undirected graph partition result can decide which points belong to which line. The steps are described as follow.

To begin with, the concept of main edge and minor edge is given in Fig. 3(a). When the edge between q_i and q_j meet the criterion that j=i+1, the edge is a main edge, otherwise it is a minor edge. At a word, the main edge is composed of the adjacent sub groups. It's accordant with the fact that main edges play a more decisive role in line segments merge than minor edges.

Then the seed edge and break edge(Fig. 3(b)) are defined as follow before constructing the energy function:

1）The seed edges are defined which have the local maximum of merge probability in the main edges:

$$\mathrm{Seed}=\{\varepsilon_i|\mathrm{sign}(F_{main}(\varepsilon_i)')'=-2\ \&\ F_{main}(\varepsilon_i)>\delta\}$$

2）The break edges are defined whose merge probability is smaller than threshold σ. Break edges represent the maximum range of the growth of seed edges.

$$\mathrm{Break}=\{\varepsilon_j|F_{main}(\varepsilon_j)<\sigma\},\varepsilon_j\in\varepsilon$$

The nodes are pre-divided into N classes by break edges. In order to segment the undirected graph, we construct an energy function of undirected graph partition as follow:

$$E(M)=\min_{M\geq N}-\log\left(\frac{\sum_{k=1}^{M}E_{inter}(S_k)}{M}\right)$$

The undirected graph is divided into M classes $\{S_k\}_{k=1}^{M}$ by discretely optimizing the energy function. $E_{inter}(S_k)$ is the energy function of inner-class which is given as:

$$E_{inter}(m_k)=\frac{\sum_{i=1}^{m_k}\sum_{j=i+1}^{m_k}\frac{F(q_i,q_j)}{j-i}}{m_k-1}$$

where m_k denotes the number of nodes in S_k; m_k-1 means the number of main edges in S_k and $j-i$ measures the weights of edges in S_k. $E_{inter}(m_k)$ is the weighted sum of edges(Fig. 3(c)) in the class. The essence is to achieve maximum probability in the class and minimum probability between the classes.

It's known that the undirected graph partition without fixed segment number is an uncertain solvable problem. Even defined perfectly of energy function it needs much time to reach the global optimum. Considering the request of real-time, we only need a local optimal solution instead of global optimal solution. So a minimization approach using seed edge and break edge is proposed listed in Approach 1.

Mathematically speaking, if we minimize the energy function without constraint, it will need much calculation. In our method the break edges promise the minimum probability between the classes and $E_{inter}(S_k)$ makes sure the maximum probability in the class respectively. We assume that the M remains unchanged, E(M) will be larger if we put a node into other class according to the step3. Also when we assume the change of M, it's found that $E_{inter}(S_k)$ and M have the same trend. Qualitatively, the smaller M means the smaller energy function. As a result, our approach can get a local minimum especially a global minimum when class number is fixed.

Physically speaking, the adjacent relationship between nodes can best represent the similarities and differences of these two nodes. However, the indirect relationship has the limit effect of auxiliary interpretation, which will determine the segmentation only if it accumulates to a certain level. That's the meaning of main edge and minor edge.

Approach 1. Energy Function Minimization Approach

Step1: Choose an edge from **Seed**, add the nodes of seed edge into S_k, grow forward or backward along main edges.
Step2: Add the new node and the corresponding edges into current S_k along the growth direction, compute $E_{inter}(S_k)$.
Step3: When faced with the following two cases, terminate the addition of new node: (1) $E_{inter}(S_k)$ gets smaller (2) reach the break edges. If not, go to **step 2**.
Step4: Set the main edge which stops growing as break edge, update the **Seed** and **Break.**
Step5: When the **Seed** and **Break** is traversed completely, end the algorithm, or go to **step1** and k=k+1.

VII. Experiments

We designed two experiments to show the performance of our approach and compared with other typical methods.

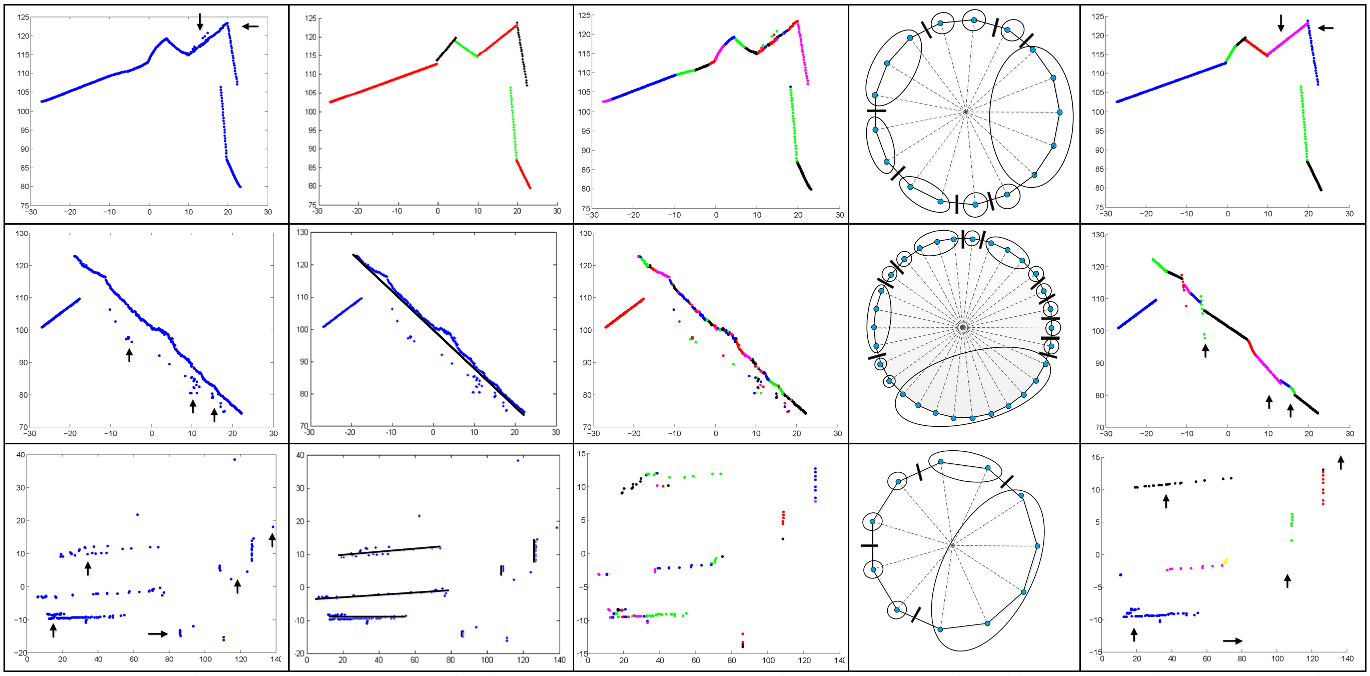
There are two laser scan datasets used in the tests. The first one is taken by laser profile sensor on robotic arm for the outer contour of workpiece in Fig. 4(a). The other is got by laser range finder on XJTU's unmanned vehicle for obstacle detection in Fig. 4(b). Though these two laser sensors have different working principle, they both move in a specific way (straight line or free-movement) and get a set of 2D scan data. In the first dataset the shape of points is simple but requires high detection precision. On the contrary, it's more complicated in the second dataset which asks for high detection accuracy.

(a) Laser profile sensor (b) Laser range finder

Fig. 4. Two kinds of sensors to gather laser scan datasets

A. *Typical Data Experiments*

In this section three laser data from two dataset is selected to evaluate the performance of our approach. The first two are taken by laser profile sensor and the last one is got by laser range finder. Each of them has unique characteristic to prove different function effect. The first scan data is standard surface contour of workpiece with local noise and global turns at the same time. The second one has abundant noise and distortion. The third scan data is vehicle radar data with a lot of outlier. All these are difficult to detect in the same parameters.



(1) Raw scan data (2) Groundtruth (3) Over segmentation (4) Graph partition (5) Line extraction results

Fig. 5. Experiment results of our method. Images from top line are numbered as a, b and c. The scan data in each line has unique characteristic to prove different function effect. Arrows in first and last column point to the difficulty of this kind of data.

In this experiment, the parameters of our approach are selected as follows: the IEPF parameters $threshold_0$ is set to 1/500 of the length of scan data; the step length λ is defined as 1/2 of $threshold_0$. Break threshold σ is set to 37.5% and Seed threshold δ is 65%. α, β and γ are 1 in general.

The results of our approach are shown in Fig. 5. As seen that our algorithm have good line extraction performance on both databases. The extracted line segments correspond to the groundtruth quite well. The scan data (a-b) got by laser profile sensor is dense. The detection result (a5) shows that our algorithm can deal with the significant turns and local small noise simultaneously in one threshold and get good effect. We can see that there are much noise and deviation in raw scan data in (b1). For this kind of data, our algorithm obtains longer line segments result instead of many tiny line segments.

The scan data (c) got by laser range finder is sparse. It contains more noise because it was collected on the road outdoor. The result in (c5) shows that we can extract line segments of structural points in noisy radar data correctly. Also jump-border of lines and large deviation points in laser scan data are well eliminated and they won't affect the line extraction of our algorithm

It shows the process of undirected graph partition in column 4 of Fig. 5. The outer edges of polygon represent main edge. The thick lines on main edge are initial break edge. The ovals mean the cluster results of seed edge growth. It can be seen that the edges between adjacent break edges are divided into fragments in different size and number by the constrain of $E_{inter}(m_k)$ to accomplish undirected graph partition.

Furthermore, the average run time of our approach is shown in Table I as follow:

TABLE I
AVERAGE RUN TIME OF OUR APPROACH (MS)

Preprocessing and over-segmentation	**Undirected graph generated (Kmeans cluster)**	**Line segments extraction**	**Total time**
1.966	7.01(6.13)	0.3	9.276

Our method was programmed in matlab. It was executed on a personal computer with intel core(TM) i7-2720 2.2GHz and 8GB RAM. Our algorithm can reach 100Hz which meets the real-time requirements. The calculation of merge probability especially kmeans cluster costs most time. Also the Energy function minimization in section 3 doesn't spend much time.

B. *Comparative Experiments*

In order to prove the correctness of our approach further, we designed a comparative experiment on former datasets. Base on the performance and popularity of existing line segments extraction algorithm on 2D laser data, we select 6 algorithms to complete the experiment. They are SEF[2], LT[3], Split and Merge[5], IEPF[8], RANSAC[14] and HT[10].

51 scan data got by laser profile sensor and 22 scan data got by laser range finder are selected in comparative experiment. The groundtruth of selected data is calibrated manually according to the actual situation of workpiece and traffic environment. The groundtruth consists of several true lines which is given by $L^T = \{L_k^T|(\kappa_k^T, \beta_k^T, x_k^{a^T}, y_k^{a^T}, x_k^{b^T}, y_k^{b^T})\}_{k=1}^{N^T}$. An extracted line can't belong to more than one true line. So a true line is correctly matched if more than 60% points of true line are included in one extracted line L_k.

In order to illustrate the simulation results, three kinds of quality measures were evaluated: speed, correctness and accuracy. The correctness measures include the recall and precision.

$$\mathrm{Recall} = \frac{\mathrm{N.Match}}{\mathrm{N.Groundtruth}}$$
$$\mathrm{Precision} = \frac{\mathrm{N.Detect} - \mathrm{N.Match}}{\mathrm{N.Detect}}$$

where $\mathrm{N.Match}$ is the number of matched lines, $\mathrm{N.Groundtruth}$ is the number of true lines in groundtruth and $\mathrm{N.Detect}$ is the number of lines extracted by an algorithm.

The accuracy measures are the evaluation criterion of matched lines defined as follow:

$$\mathbf{PointError} = \frac{\text{Error detection points number}}{\text{Total points number}}$$
$$\mathbf{EdgeError} = \frac{||\left(x_k^{a^T}, y_k^{a^T}\right) - (x_k^a, y_k^a)|| + ||\left(x_k^{b^T}, y_k^{b^T}\right) - (x_k^b, y_k^b)||}{||\left(x_k^{b^T}, y_k^{b^T}\right) - \left(x_k^{a^T}, y_k^{a^T}\right)||}$$
$$\Delta\kappa = \kappa_k - \kappa_k^T, k = 1, \dots, N$$
$$\Delta\beta = \beta_k - \beta_k^T, k = 1, \dots, N$$

where $\Delta\kappa$ and $\Delta\beta$ are the mean absolute error of matched line parameters. **EdgeError** and **PointError** describe the accuracy of matched lines' edge and body to true lines.

Since most algorithm performances are very sensitive to the threshold, choosing parameter values is an important task. In our comparison experiment we do a lot of work to adjust the parameters of each algorithm, in order to make sure their best performance on datasets.

TABLE II
COMPARISON RESULTS ON DATASETS

Algorithm	Speed (Hz)	Correctness		Accuracy			
		Recall	Precision	PointError	EdgeError	Δκ	Δβ
SEF	384.1	0.5495	0.5222	0.0946	0.1247	0.091	1.718
LT	78.09	0.7289	0.5821	0.0977	0.1271	0.069	1.052
Split-Merge	112.5	0.6484	0.6855	0.0994	0.1196	0.156	2.924
IEPF	**543.8**	0.8136	0.6893	0.0766	0.0854	0.053	0.9226
RANSAC	12.95	0.4848	0.5685	0.1291	0.4307	**0.038**	**0.5604**
HT	25.96	0.3212	0.2576	0.1645	0.2181	0.300	6.9155
Ours	138.8	**0.8474**	**0.7740**	**0.0424**	**0.0557**	0.042	0.7559

The comparative results on datasets are show in Table II. The bold means the best performance for each quality measure. IEPF is the fastest method because of its simple iteration strategy. Our method has middle speed which meets real-time requirements. Furthermore, our method has highest value in recall and precision, which indicates the validity and superiority. HT is the lowest in correctness and accuracy for the truncation errors and distortion when transforming into image. RANSAC has bad result in PointError and EdgeError because it doesn't follow the continuity of points. Also it has the best result in $\Delta\kappa$ and $\Delta\beta$ for the same reason. However it's low rate in correctness indicates that it mistakenly identified many line segments. Our method has best result in PointError and EdgeError and ranks second in $\Delta\kappa$ and $\Delta\beta$.

VIII. CONCLUSION

We have proposed a novel line segments extraction algorithm of 2D laser scan data using undirected graph. The concept of merge probability, main edge, seed edge and break edge were defined. Also the energy function of graph partition and its minimization approach were put forwarded to extract line segments. The primary contributions of the work are the main edge, undirected graph and minimization method of energy function, which are more proper to noisy laser scan data. The typical experiment results show that our method has a good performance on two kind typical laser data. The comparison to many stat-of-the-art techniques on challenging data indicates the superiority of our approach.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (NSFC) under Grant No.61305051 and Grant No.61328303.

REFERENCES

[1] G. A. Borges, M. J. Aldon, "Line extraction in 2Ds range images for mobile robotics," Journal of Intelligent and Robotic Systems, vol. 40, no. 3, pp. 267-297, 2004.

[2] A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R. Husson, "An optimized segmentation method for a 2D laserscanner applied to mobile robot navigation," in Proc. SICICA'97, Annecy Le Vieux, France, 1997, pp.153-158.

[3] 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 Proc. IEEE ICRA'96, Minneapolis, USA, 1996, pp. 901-908.

[4] K. Georgiev, R. T. Creed, R. Lakaemper, "Fast plane extraction in 3D range data based on line segments," in Proc. IEEE IROS'11, San Francisco, 2011, pp. 3808-3815.

[5] T. Pavlidis, S. L. Horowitz, "Segmentation of Plane Curves," IEEE Trans. on Computers, vol. c-23, no. 31, pp.860-870, 1974.

[6] B. Sarkar, P. K. Pal, D. Sarkar, "Building maps of indoor environments by merging line segments extracted from registered laser range scans," Robotics and Autonomous Systems, vol. 62, no.4, pp. 603-615, 2014.

[7] G. A. Borges, M. J.Aldon, "A split-and-merge segmentation algorithm for line extraction in 2d range images," in Proc. ICPR'00, Barcelona, Spain, 2000, vol. 1, pp. 441-444.

[8] T. Einsele, "Real-time self-localization in unknown indoor environment using a panorama laser range finder," in Proc. IROS '97, Grenoble, France, 1997, vol. 2, pp. 697-702.

[9] D. A. Forsyth, J. Ponce, "Computer vision: a modern approach," New York, Prentice Hall, 2003.

[10] M. Ogaz, R. Sandoval, M. Chacon, "Data processing from a Laser Range Finder sensor for the construction of geometric maps of an indoor environment," in Proc. IEEE MWSCAS'09, Cancun, Mexico, 2009, pp. 306-313.

[11] Z. Xu, B. S. Shin, R. Klette, "Accurate and Robust Line Segment Extraction Using Minimum Entropy with Hough Transform," IEEE Trans. On Image Processing, vol. 24, no. 3, pp. 813-822, 2014.

[12] G. von, J. Jakubowicz, J. M. Morel, "LSD: A Fast Line Segment Detector with a False Detection Control," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 722-732, 2008.

[13] Y. Lin, C. Wang, J. Cheng, et al. "Line segment extraction for large scale unorganized point clouds," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 102, pp. 172-183, 2015.

[14] V. Nguyen, S. Gächter, A. Martinelli, et al. "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics," Autonomous Robots, vol. 23, no. 2, pp. 97-111, 2007.

[15] J. Vandorpe, H. Van Brussel, H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder," in Proc. IEEE ICRA'96, Minneapolis, USA, 1996, vol. 1, pp. 901-908.

[16] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino, "Mobile robot SLAM for line-based environment representation," in Proc. IEEE CCD-ECC'05, Seville, Spain, 2005, pp. 2041-2046.

[17] Y. H. Choi, T. K. Lee, S. Y. Oh, "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot," Autonomous Robots, vol. 24, no. 1, pp. 13-27, 2008.