A Split-and-Merge Segmentation Algorithm for Line Extraction in 2-D Range Images

G. A. Borges * M.-J. Aldon

Robotics Department, LIRMM, UMR CNRS/Université Montpellier II, n^o. C55060
161, rue ADA. 34392 - Montpellier - Cedex 5 - France
E-mail: {borges,aldon}@lirmm.fr

Abstract

This paper presents a segmentation method for line extraction in 2-D range images. It uses a prototype-based fuzzy clustering algorithm in a split-and-merge framework. The split-and-merge structure allows us to use the fuzzy clustering algorithm without any previous knowledge on the number of prototypes. This algorithm aims to be used in mobile robots navigation systems for dynamic map building. Simulation results show its good performance compared to some classical approaches.

1. Introduction

Environment maps are extensively used in mobile robots navigation systems for tasks like map-based positioning or global path planning. When the robot moves on a flat ground, a 2-D map is sufficient to solve these problems. For indoor environments, a such map can be extracted from a set of range data provided by a rotating laser range finder. Planar surfaces that often occur in structured environments are modeled by line segments. The process of line extraction from range data must be executed on line while the robot is moving, and it must provide an accurate polygonal model of the environment. Classical algorithms initially used for this task [3, 6, 7] are issued from the edge segmentation methods developed for video image processing. These algorithms are generally very sensitive to changes on some parameters. More sophisticated methods like prototype-based fuzzy clustering algorithms [1, 5] are more robust but they are generally more time expensive. In order to avoid some drawbacks of these solutions, this paper proposes a split-and-merge segmentation algorithm based on the fuzzy clustering approach.

In this paper, Section 2 presents some classical line extraction algorithms as well as the fuzzy clustering approach with a new representation for the line parameters. The proposed algorithm is described in Section 3 and compared with other approaches in the simulation experiment in Section 4. The conclusions of this research are presented in Section 5.

* Work supported by a grant from Capes - Brasília, Brazil.

2. Line extraction algorithms

A 2-D range image is represented by $\mathcal{Z} = \{(x_j, y_j) \mid j = 1, \ldots, N\}$, where (x_j, y_j) are the cartesian coordinates of the *j-th* point. The points are sequentially acquired by the laser range finder with a given angular resolution.

2.1. Sequential algorithms

Some algorithms as the successive edge following (SEF) and the line tracking (LT) process the data points sequentially [6, 7]. The SEF terminates a line segment on the point (x_{n-1}, y_{n-1}) if its distance to the next point (x_n, y_n) is greater than a threshold $D_{\max}$. The search for a new line is started from (x_n, y_n). The LT algorithm verifies the distance T_n of the current point (x_n, y_n) to the line fit by the accumulated last support points $\mathbf{p} = \{(x_{n-1}, y_{n-1}), (x_{n-2}, y_{n-2}), \ldots, (x_i, y_i)\}$. If $T_n > T_{\max}$ the already fit line is stored in a list and a new line search starts from (x_{n+2}, y_{n+2}) with $\mathbf{p} = \{(x_{n+1}, y_{n+1}), (x_n, y_n)\}$. These algorithms suffer from a high sensibility to the thresholds $D_{\max}$ and $T_{\max}$.

2.2. Recursive algorithms

A well known recursive algorithm is the iterative end point fit (IEPF) [3]. This algorithm recursively splits a set of points $\mathbf{p} = \{(x_N, y_N), (x_{N-1}, y_{N-1}), \ldots, (x_1, y_1)\}$ into two subsets $\mathbf{p}' = \{(x_N, y_N), ..., (x_a, y_a)\}$ and $\mathbf{p}'' = \{(x_a, y_a), ..., (x_1, y_1)\}$ if a validation criterion is not satisfied. $(x_a, y_a) \in \mathbf{p}$ is the point for which the distance T_a to the line formed by the extreme points of $\mathbf{p}$ is maximum.

The criterion validates lines with $T_a \leq T_{\max}$. This is a recursive algorithm because the same procedure is repeated to $\mathbf{p}'$ and $\mathbf{p}''$ until all validation criteria are satisfied. Due to its good performance and being computationaly inexpensive, the IEPF and its derivatives are commonly used for line extraction in range images [4].

2.3. Prototype-based fuzzy clustering algorithm

The main objective of prototype-based fuzzy clustering algorithms is to reduce iteratively a cost function J. Most algorithms use a cost function given by

$$J(\beta,\mathcal{U},\mathcal{Z}) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^m d^2(x_j,y_j,\beta_i). \quad (1)$$

In eq. (1), β_i represents the parameters of the *i-th* prototype and u_{ij}^m is the grade of membership of the *j-th* point (x_j,y_j) to the prototype β_i. $d(x_j,y_j,\beta_i)$ is a distance function between the point (x_j,y_j) and the prototype β_i and m is a constant. The fuzzy clustering algorithm also imposes the following constraints:

$$u_{ij} \in [0,1], \quad 0 < \sum_{j=1}^{N} u_{ij} < N \ \text{ for all } i, \quad (2)$$

$$\sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j, \quad (3)$$

For the purposes of straight lines extraction, there exist different representations for the prototypes β_i based on the cluster covariance matrix [1, 5]. However, the new approach proposed in this paper uses a more compact representation. This representation is given by $\beta = \{(\rho_i,\alpha_i) \mid i = 1,\ldots,C\}$, where $\beta_i = (\rho_i,\alpha_i)$ are the polar parameters of the *i-th* straight line. From the polar representation of a line $\rho = x\cos(\alpha) + y\sin(\alpha)$ [8], a suitable choice for the distance function d is

$$d^2(x_j,y_j,\beta_i) = f^2(x_j,y_j,\beta_i) + g^2(x_j,y_j,\beta_i), \quad (4)$$

where $f^2(x_j,y_j,\beta_i) = (\rho_i - x_j\cos(\alpha_i) - y_j\sin(\alpha_i))^2$, and $g^2(x_j,y_j,\beta_i)$ is a penalty function for points that are very far from the cluster center.

Partitioning the data set $\mathcal{Z}$ into C prototypes β is accomplished by minimizing the objective function J (eq. (1)). In order to consider the constraints (3), the Lagrange multipliers method is used and the problem becomes to minimize

$$V(\beta,\mathcal{U},\mathcal{Z}) = J(\beta,\mathcal{U},\mathcal{Z}) + \sum_{j=1}^{N}\lambda_j\left(1 - \sum_{i=1}^{C} u_{ij}\right), \quad (5)$$

where the λ_j's are the Lagrange's multipliers. V is minimized by using the alternating optimization method. Therefore, by considering β as constant, u_{ij} is determined as the one that satisfies $\partial V(\beta,\mathcal{U},\mathcal{Z})/\partial u_{ij} = 0$ and $\partial V(\beta,\mathcal{U},\mathcal{Z})/\partial\lambda_j = 0$. Thus, u_{ij} is given by

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left[\frac{d^2(x_j,y_j,\beta_i)}{d^2(x_j,y_j,\beta_k)}\right]^{\frac{1}{m-1}}}. \quad (6)$$

In order to determine the parameters of the prototype β_i, the membership measures u_{ij} are considered as constants and the following equations set is solved: $\partial J(\beta,\mathcal{U},\mathcal{Z})/\partial\beta_i = \mathbf{0}$. Therefore, using eqs. (1) and (4), it can be shown that ρ_i and α_i are given by

$$\rho_i = \tilde{x}_i\cos(\alpha_i) + \tilde{y}_i\sin(\alpha_i), \quad (7)$$

$$\alpha_i = \frac{1}{2}\arctan\left(\frac{-2\tilde{S}_{xy_i}}{\tilde{S}_{yy_i} - \tilde{S}_{xx_i}}\right), \quad (8)$$

where

$$\tilde{x}_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m}, \quad \tilde{y}_i = \frac{\sum_{j=1}^{N} u_{ij}^m y_j}{\sum_{j=1}^{N} u_{ij}^m}, \quad (9)$$

$$\tilde{S}_{xx_i} = \sum_{j=1}^{N} u_{ij}^m (x_j - \tilde{x}_i)^2, \quad \tilde{S}_{yy_i} = \sum_{j=1}^{N} u_{ij}^m (y_j - \tilde{y}_i)^2,$$

$$\tilde{S}_{xy_i} = \sum_{j=1}^{N} u_{ij}^m (x_j - \tilde{x}_i)(y_j - \tilde{y}_i).$$

In the above development, the penalty function g was kept as constant and given by

$$g^2(x_j,y_j,\beta_i) = (x_j - \tilde{x}_i)^2 + (y_j - \tilde{y}_i)^2, \quad (10)$$

where $\tilde{x}_i$ and $\tilde{y}_i$ are the weighted center of gravity coordinates of the *i-th* prototype from the last iteration (eq. (9)).

The prototype-based fuzzy line extraction algorithm is summarized as follows:

1. There are given the data set $\mathcal{Z}$, the number of lines C and the initial prototypes β;
2. Update the membership grade measures u_{ij}, for $i = 1,\ldots,C$ and $j = 1,\ldots,N$, using eq. (6);
3. Update the prototype parameters β_i for $i = 1,\ldots,C$, using eqs. (7) and (8);
4. If for at least one prototype parameter its change is greater than a given threshold, go to step 2;
5. End.

The main drawbacks of this algorithm are its sensitivity to initialization (e.g. the initial prototypes), to local minima of J, to noise and outliers and the difficulty to determine *a priori* the number of lines C. There exist other approaches to deal with these drawbacks as shown in [2].

3. The proposed algorithm

The algorithm SMF (split-and-merge fuzzy) proposed in this paper uses the previous fuzzy clustering approach presented in Section 2.3 in a split-and-merge framework. In the SMF, a list of lines $\mathcal{L} = \{\mathbf{l}_k \mid k = 1, \ldots, N_L\}$ is iteratively updated. In this list the *k-th* line is represented by $\mathbf{l}_k = (\rho_k, \alpha_k, \bar{x}_k, \bar{y}_k, \mathbf{q}_k)$, where ρ_k and α_k are its polar parameters and $(\bar{x}_k, \bar{y}_k)$ are the coordinates of the center of gravity of the set of points that generated this line. Such points are also called support points, and their index *j* are stored in the vector $\mathbf{q}_k$. During the split phase, the lines are generated following a validation criterion based on the dispersion of their support points given by

$$\sigma_k^2 = \frac{1}{s_k} \sum_{j \,\in\, \mathbf{q}_k} d^2(x_j, y_j, \beta_k), \tag{11}$$

where s_k is the number of support points.

The split phase corresponds to the following procedure:

1. Initialization:
   (a) $\mathcal{L}$ is initially composed of just one line: the one that best fit all points $\mathcal{Z}$, and $N_L^{(0)} = 1$.
   (b) Set the generation number $m = 1$;
2. Update the *m-th* generation list size as $N_L^{(m)} := N_L^{(m-1)}$;
3. For each line $\mathbf{l}_k$, $k = 1, \ldots, N_L^{(m-1)}$, in $\mathcal{L}$ that does not satisfy $\sigma_k \leq \sigma_{\max}$, do:
   (a) Create two new lines $\mathbf{l}'_k$ and $\mathbf{l}''_k$ from the support points of $\mathbf{l}_k$ by executing the fuzzy line extraction algorithm with $C = 2$;
   (b) Substitute $\mathbf{l}_k$ by $\mathbf{l}'_k$ and add $\mathbf{l}''_k$ to the end of $\mathcal{L}$.
   (c) $N_L^{(m)} := N_L^{(m)} + 1$;
4. If there is at least one line $\mathbf{l}_k$, $k = 1, \ldots, N_L^{(m)}$, that does not satisfy $\sigma_k \leq \sigma_{\max}$, go to step 2;
5. End.

The only user supplied parameter is the threshold $\sigma_{\max}$. This parameter may be determined given a statistical analysis of the range sensor errors.

In the merge phase, for each line $\mathbf{l}_k$ in $\mathcal{L}$ two other lines $\mathbf{l}_a$ and $\mathbf{l}_b$ are chosen as fusion candidates. These candidates are the two closest lines to $\mathbf{l}_k$ given the distance of their center of gravity coordinates. In this work, the fusion of two lines $\mathbf{l}'$ and $\mathbf{l}''$ is represented by $\mathbf{l} = \mathbf{l}' \hat{\ } \mathbf{l}''$ and is defined as the line $\mathbf{l}$ fit by the support points of $\mathbf{l}'$ and $\mathbf{l}''$. The fused line $\mathbf{l}_f$ is the one that gives the smallest dispersion between $\mathbf{l}_k \hat{\ } \mathbf{l}_a$ and $\mathbf{l}_k \hat{\ } \mathbf{l}_b$ and that also satisfies the dispersion criterion $\sigma_f \leq \sigma_{\max}$. The merge phase exits when there is no more possible fusions.

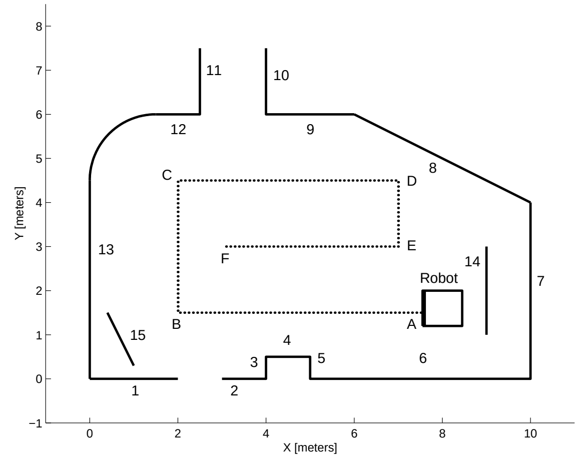

Figure 1. Simulation environment.

4. Evaluation

In order to compare the SMF algorithm and the classical LT and IEPF algorithms, experiments were carried out in the simulated environment shown in Figure 1. Due to the bad results of the SEF algorithm in previous simulations, this one was not used in this experiment.

In this simulation, a mobile robot equipped with a laser range finder moves among the obstacles represented by the lines numbered from *1* to *15* (Fig. 1). The range finder has an angular field of view of 270^o and a resolution of 0.6^o. The measured distances are contaminated with a Gaussian random noise whose the standard deviation is 5.0 cm. These sensor parameters are the same as those of the *Ladar 2D30 IBEO Lasertechnik.*

The experiment consisted in moving the robot following the trajectory *ABCDEF* and generating 200 synthetic range images at intermediary positions on this trajectory. For each range image, the three algorithms LT, IEPF and SMF were executed. In order to illustrate the simulation results, four quality measures were evaluated. These measures are statistics related to the lines extracted from each environment line $\mathbf{e}_l = (\rho_l^e, \alpha_l^e)$, $l = 1, \ldots, 15$. Let $\mathcal{L}_l \in \mathcal{L}$ be a sub-list of N_l extracted lines with support points generated by $\mathbf{e}_l$. An extracted line cannot belong to more than one set $\mathcal{L}_l$. However a given extracted line $\mathbf{l}_k$ may have support points generated from different environment lines. In this case, $\mathbf{l}_k$ is associated with the sub-list of the more similar environment line in the polar parameters sense. As a given environment line $\mathbf{e}_l$ may be associated to one or more extracted lines as consequence of the algorithm behavior, it is interesting to measure the validity of such extracted lines. In doing so, a given line $\mathbf{l}_k$ extracted from an environment line $\mathbf{e}_l$ is valid iff $|\rho_l^e - \rho_k| \leq 0.5\ m$ and $|\alpha_l^e - \alpha_k| \leq 20^o$. Therefore, M_l being the number of valid extracted lines from the sub-

Table 1. Results on $\bar{N}_l$ and $\bar{\eta}_l$.

Environment	$\bar{N}_l$			$\bar{\eta}_l$		
line e_l	LT	IEPF	SMF	LT	IEPF	SMF
1	1.28	1.85	1.01*	0.86	0.80	0.88*
2	1.81	2.58	1.14*	0.64	0.41	0.84*
3	1.07	1.15	1.00*	0.07	0.32*	0.14
4	1.91	2.65	1.20*	0.66	0.57	0.88*
5	1.11	1.21	1.00*	0.72*	0.51	0.56
6	3.88	6.64	1.13*	0.62	0.45	0.94*
7	1.98	2.48	1.47*	0.39	0.33	0.42*
8	3.66	6.23	1.19*	0.63	0.45	0.93*
9	2.61	4.17	1.13*	0.61	0.46	0.94*
10	1.12	1.15	1.03*	0.46	0.50	0.60*
11	1.08	1.07	1.00*	0.25	0.51*	0.33
12	1.79	2.21	1.11*	0.76	0.74	0.94*
13	4.07	6.46	1.48*	0.28*	0.18	0.27
14	1.82	2.72	1.07*	0.50	0.48	0.56*
15	2.32	3.04	1.34*	0.70	0.57	0.89*

Table 2. Results on $\bar{\varepsilon}_\rho$ and $\bar{\varepsilon}_\alpha$.

Environment	$\bar{\varepsilon}_\rho$ in cm			$\bar{\varepsilon}_\alpha$ in degrees		
line e_l	LT	IEPF	SMF	LT	IEPF	SMF
1	6.13	9.25	2.85*	2.57	3.83	1.05*
2	11.75	11.35	8.12*	2.00	2.07	1.42*
3	4.14	2.08	1.41*	8.32	4.98*	6.31
4	18.85	18.18	13.85*	4.18	5.23	2.11*
5	5.42	3.45*	4.34	9.36	6.60*	8.43
6	13.97	19.09	3.88*	2.20	4.65	0.32*
7	19.46	20.38	10.48*	5.07	6.51	2.66*
8	12.89	17.68	4.79*	2.59	4.07	0.76*
9	15.79	19.60	6.74*	2.25	3.70	0.82*
10	9.17*	13.52	14.35	0.77*	1.99	1.19
11	10.71*	11.18	12.76	0.97*	1.19	1.10
12	15.70	14.08	8.01*	4.82	5.06	2.19*
13	16.50	20.77	4.16*	3.75	4.36	1.00*
14	12.28	14.50	3.47*	4.84	7.29	1.19*
15	6.70	7.11	2.46*	6.47	7.25	2.31*

list $\mathcal{L}_l$, let define the validation ratio $\eta_l = M_l/N_l$, with $0 \leq \eta_l \leq 1$. η_l is close to 1 when the extracted lines look like the corresponding environment line $\mathbf{e}_l$.

The parameters employed in this simulation were $T_{\max} = 10\ cm$ for the LT and IEPF algorithms, and $\sigma_{\max} = 5\ cm$ for the SMF algorithm. Table 1 presents the mean values $\bar{N}_l$ and $\bar{\eta}_l$ of N_l and η_l given the extracted lines obtained from the LT, IEPF and SMF algorithms on the 200 range images. The values followed by * indicate the best performances. The SMF algorithm gave lower values of $\bar{N}_l$ for all environment lines than the LT and IEPF algorithms. This indicates that the SMF algorithm breaks less environment lines than the others ones. For $\bar{\eta}_l$, the SMF also presented the best results, except for the environment lines *3*, *5*, *11* and *13*. Table 2 presents the mean absolute errors $\bar{\varepsilon}_\rho$ and $\bar{\varepsilon}_\alpha$ on the polar parameters ρ_k and α_k of the extracted lines. These measures are referred to the corresponding environment line $\mathbf{e}_l$. The SMF also gave lower mean absolute errors except for the environment lines *3*, *5*, *10* and *11*. Therefore, for these lines the differences with respect to the LT and IEPF algorithms are very small: less than *6 cm* for ρ_k, and less than 2^o for α_k. For all the other lines, the SMF gave the best results with discrepancies in $\bar{\varepsilon}_\rho$ and $\bar{\varepsilon}_\alpha$ that are not negligible for some lines.

5. Conclusion

This paper describes a new line extraction algorithm for 2-D range images. It uses a prototype-based fuzzy clustering algorithm in a split-and-merge framework. The split-and-merge structure allows the use of the fuzzy clustering algorithm without previous knowledge of the number of prototypes. Therefore, other more robust fuzzy clustering algorithms me be also used. Experiments carried out on a simulated environment showed the good performance of the SMF compared with other most used approaches. This method could be applied for line extraction in video images.

References

[1] J. C. Bezdek. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York, USA, 1981.

[2] R. N. Davé and R. Krishnapuram. Robust Clustering Methods: A Unified View. *IEEE Transactions on Fuzzy Systems*, 5(7):270–293, May 1997.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, USA, 1973.

[4] T. Einsele. Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder. In *Proceedings of IROS*, pages 697–702, 1997.

[5] H. Frigui and R. Krishnapuram. A Robust Competitive Clustering Algorithm With Applications in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, May 1999.

[6] A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R. Husson. An Optimized Segmentation Method for a 2d Laser-scanner Applied to Mobile Robot Navigation. In *3rd IFAC Symp. on Intelligent Components and Instruments for Control Applications*, pages 153–158, June 9-11 1997.

[7] 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 of the IEEE Int. Conf. on Robotics and Automation*, pages 901–908, April 1996.

[8] T. Y. Young and K.-S. Fu. *Handbook of Pattern Recognition and Image Processing*. Academic Press, Inc., London, UK, 1986.