Proceedings of the 6th International Conference on Automation, Robotics and Applications, Feb 17-19, 2015, Queenstown, New Zealand

A Connected Component Labeling Algorithm for Sparse Lidar Data Segmentation

Abhijeet Ravankar, Yukinori Kobayashi, Ankit Ravankar, Takanori Emaru
Division of Human Mechanical Systems and Design
Graduate School of Engineering, Hokkaido University, Sapporo, Japan
Email: abhijeet@ec.hokudai.ac.jp, kobay@eng.hokudai.ac.jp

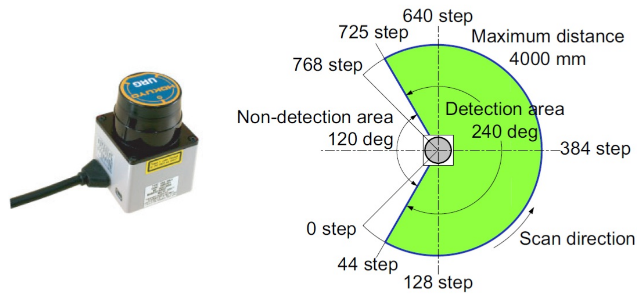
***Abstract*—This paper proposes an extended connected-components labeling algorithm for sparse Lidar (Light detection and ranging) sensor data. It is difficult to label sparse Lidar data using the general connected-component labeling algorithm. The proposed technique first increases the density of the sparse data by performing mathematical morphological operation of dilation. Next, labeling is performed on the dilated data, and the resultant labels are mapped to the input sparse Lidar data. The proposed technique does not distort the input Lidar data. We show the application of the proposed algorithm in map building using clustering. Results show that the proposed method can label sparse Lidar data to build maps.**

I. Introduction

For a mobile robot to autonomously navigate in an unknown environment, building accurate maps is very essential. In case of rescue robots and indoor navigation, small open spaces like open doors, windows, and narrow passages needs to be accurately mapped in order to safely and accurately navigate the robot. Most mobile robots perceive the outer world using external sensors like laser sensors, camera, sonar, bump sensors, etc. From these sensors, information pertaining to geometric features (lines, corners, walls, etc.) are used to build 2D or 3D map of the environment. In order to extract the features from the environment, data from the sensor needs to be processed. Clustering is one of the techniques which is widely used for information extraction from Lidar sensors. Clustering is a technique used in the data analysis by which large sets of data can be grouped into clusters of smaller sets of similar and meaningful data. It is one of the main techniques used in data mining, and analysis of statistical data in fields like machine learning, image processing, and pattern analysis. Various approaches to solve the mapping problem using clustering have been proposed before [1], [2].

Generally, when clustering Lidar data, noise present in the data can produce wrong clusters and miss smaller gaps (due to open doors, windows, railing gaps). To solve this problem, segmenting and accurate clustering is necessary. Connected component labeling [3], [4] is one such technique which is widely employed in image processing for blog extraction, region acquisition and data segmentation. Lidar data segmentation has been proposed in [5], [6]. However, in case of certain laser range sensors (like the widely used URG-04LX sensor), the measured data, particularly corresponding to distant objects, is sparse and it is difficult to employ component labeling directly.

This paper proposes an algorithm which extends the general connected-component labeling algorithm to label sparse 2D



(a) URG–04LX (b) Mesurement range

Fig. 1: Picture of laser range sensor URG-04LX (HOKUYO AUTOMATIC CO., LTD.)

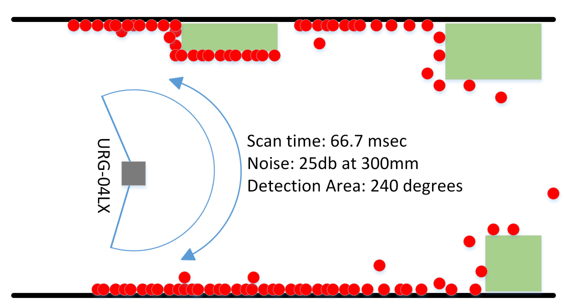


Fig. 2: URG-04LX sensor generating denser points for nearby objects and relatively sparse data for farther objects in single scan of 240 degrees.

Lidar data. This problem is different from filling out the missing information from data which occur as Lidar sensor cannot penetrate rigid objects and a ‘shadow’ appears behind those objects. Inpainting technique [7], [8] has been proposed [9] to solve this problem. This paper focuses on labeling sparse data, for which, first, mathematical morphological operation of dilation [10] is applied on sparse Lidar data. Mathematical morphology is generally used in the processing of digital images [3], [11]. Dilation increases the density of the input data to make it suitable for labeling. Next, labeling is performed on segmented data and the result is mapped with the input raw Lidar data, to obtain correct labels and data segments. Later, as an application of the proposed algorithm, we show how each segment is separately clustered, and their centroids are joined to build maps.

978-1-4799-6466-6/15/$31.00 ©2015 IEEE

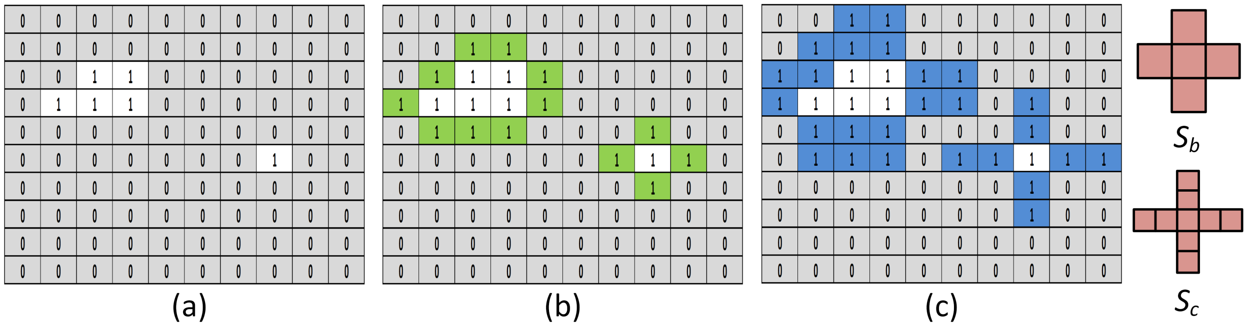


Fig. 3: Example of dilate function. Fig.(a) shows the input binary data. Fig.(b) shows the result of dilate function when structuring element S_b is used. The original data is shown in white, whereas the data added as a result of dilation is shown in green color. Fig.(c) shows the result of dilate function when structuring element S_c is used. The original data is shown in white, whereas the data added as a result of dilation is shown in blue color.

II. System specifications

Lidar sensor manufactured by Hokuyo Co. Ltd [12] is used in the experiments. Because of it's small size and light weight, it is widely used in mobile robotics research, especially for localization and mapping. The model used is Hokuyo URG-04LX shown in Figure 1. The specifications of the laser sensor have been summarized in Table I.

Widely used laser range sensors like URG-04LX can generate sparse measured data particularly when the target object is far from the sensor. Figure 2 shows a typical scan of Lidar sensor in a cluttered corridor environment. In the case of URG-04LX sensor, the noise level is around 25db at a distance of 300mm. The sensor takes only 66.7 msec to scan 240 degrees and generates 760 data points [13]. As shown in Figure 2, the sensor generates denser data around the walls which are nearer to it, whereas relatively sparse data with noise for far off targets.

The computational environment is a 64 bit Linux operating system. C++ was used to collect the data from the sensor and it was translated into 2D coordinates. The proposed algorithm was implemented in Java language.

III. Connected Component Labeling on Sparse Lidar Data

Labeling of a binary image, originally proposed by [14], is a labeled image in which the value of each element (pixel) is equal to the label of the connected component. A component labeling algorithm finds all connected components in an image (usually binary) and assigns a unique label to all points in the same component. Two data points, P and Q are said to be connected if there exists a path of points $(p_0, p_1 \cdots p_n)$ such that $p_0 = P, p_n = Q$ and $\forall 1 \leq i \leq n$, p_{i-1} and p_i are neighbors [15]. Connectivity could be 4-connected or 8-connected [3], [15]. Details of labeling can be found in various literature viz. [3], [4], [16].

TABLE I: Laser range sensor specifications

Laser range sensor	URG-04LX
Distance range	60 mm $\sim$ 4000 mm
Accuracy	20 $\sim$ 1000 mm: $\pm$25 mm
	1000 $\sim$ 4000 mm: $\pm$2.5%
Distance resolution	3.8 mm ((4000-60)mm / 1024)
Scan angle	240 deg
Angular resolution	1.875 deg (240 deg / 128)

In order to perform labeling on Lidar data, we treat the entire Lidar sensor data of the environment as a binary image of size $(W \times H)$. Here, W and H represents the width and height of the data, respectively. Each data point $D(x, y)$ where Lidar sensor has a value is treated as a *HIGH* pixel $P(x, y)$ of the image. Points where there is no Lidar sensor data is treated as a *LOW* pixel of the image. Here, *HIGH* and *LOW* represents pixel values 1 and 0, respectively. Thus, a binary representation of the Lidar data is obtained.

$$P(x_i, y_j) = \begin{cases} 1, & \text{if } D(x_i, y_j) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\forall i \in (0 \cdots W), \forall j \in (0 \cdots H)$$

Generally, component connected labeling algorithm can be applied on densely connected data. Hence, sparse Lidar data needs to be pre-processed before labeling can be applied to it. It is required to increase the density of the input data so that labeling can be applied. Since, the input Lidar data has been converted into a binary image, this is achieved by applying the mathematical morphological operation of dilation [3] on the sparse Lidar data. Dilate function is defined by [10],

$$A \oplus B = \bigcup_{b \in B} A_b \tag{2}$$

where, A represents the Lidar data probed with structuring element denoted by B. A structuring element defines the arbitrary neighborhood and can have any shape and size. Figure 3 shows how dilate function works on binary data. Figure 3(a) shows the input binary data. Figure 3(b) shows the result of dilation using S_b as the structuring element, whereas Figure 3(c) shows the result of dilation using S_c as the structuring element. Dilate operation basically looks at the neighboring elements and performs a logical *OR* operation to set the value of the current element. In other words, for

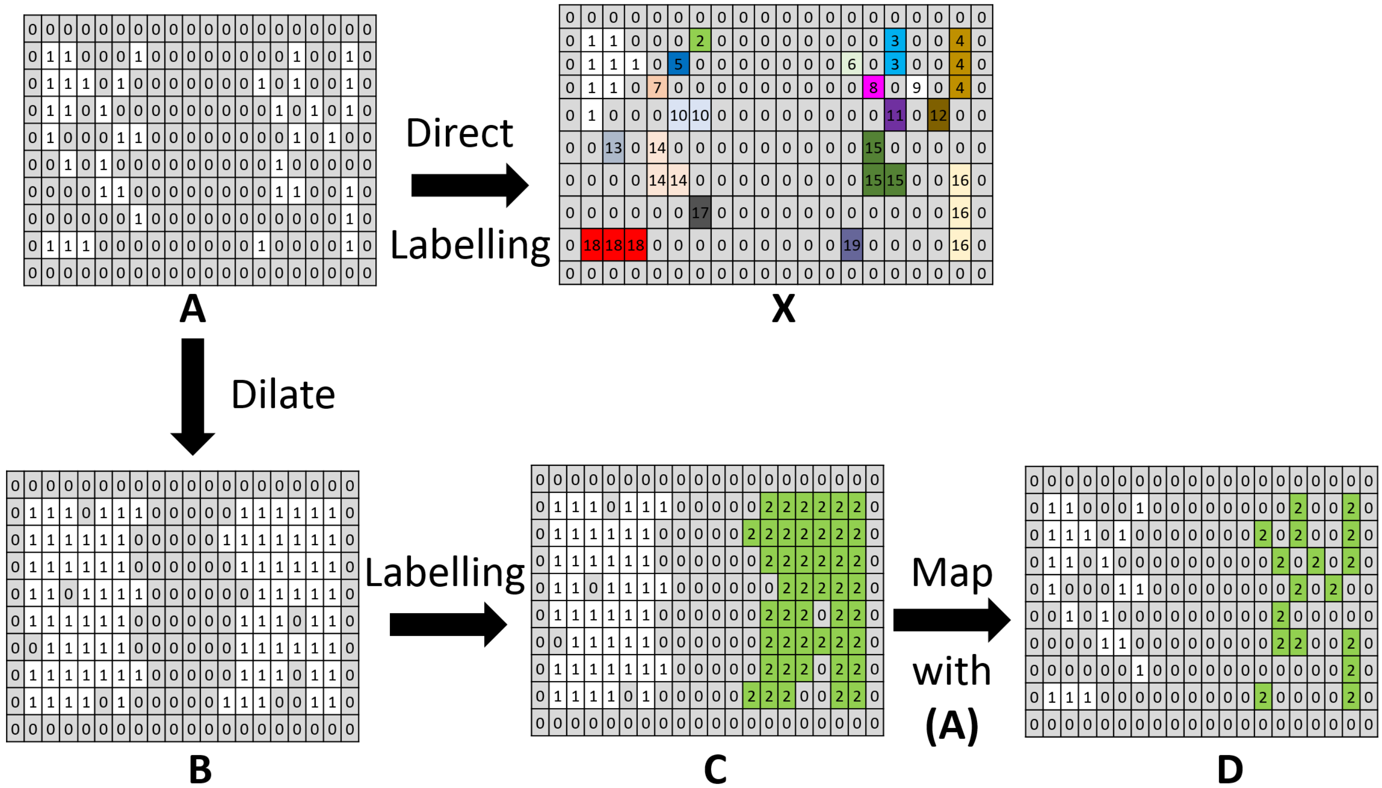


Fig. 4: Labeling of sparse Lidar data. (A) shows the sparse Lidar data. (B) shows the result of applying dilate operation on sparse Lidar data. (C) shows the result of labeling on dilated data. (D) shows the result of mapping elements of (C) with (A). (X) shows the result of directly applying labeling on sparse data. (4-connected labeling has been applied)

structuring element S_b shown in Figure 3, if $P(i,j)$ represents the input binary pixel, and $D(i,j)$ represents the dilated pixel with i^{th} row and j^{th} column, respectively, then dilate function is given by,

$$D(i,j) = P(i,j) \;\vee\; P(i-1,j) \;\vee\; P(i+1,j) \\ \vee\; P(i,j-1) \;\vee\; P(i,j+1) \tag{3}$$

Multiple dilates can be employed, or denser structuring elements can be employed to get denser data. Labeling is then performed on the dilated data and labels are obtained. Later, these labels are mapped to the corresponding points of the input data set. This process is graphically explained in Figure 4. Figure 4(A) represents the binary representation of sparse input data from Lidar sensor. Figure 4(X) shows the result of labeling (4-connected in this case) directly applied to this data. It can be seen that, since the input data is sparse, many labels (total 19 in the example) are obtained and data cannot be segmented correctly. Hence, we see that pre-processing the sparse data is essential before labeling can be applied. Dilate operation is performed on the input data and the result is shown in Figure 4(B). Dilate operation increases the density of the data. Next, labeling is performed on this data and the result is shown in Figure 4(C). Finally, all the labels of labeled data Figure 4(C) are mapped with the input data and the labels of input data are updated as shown in Figure 4(D). If $L_{dilated}(i,j)$ and $L_{input}(i,j)$ represents the label of the dilated data label (Figure 4(C)) and label of the input data (Figure 4(A)), respectively for i^{th} row and j^{th} column, then this mapping is given by,

$$L_{input}(i,j) = \begin{cases} L_{dilated}(i,j), & \forall D(i,j) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$
$$\forall i \in (0 \cdots W), \forall j \in (0 \cdots H)$$

The dilate function does not distort the actual measurements as the labeled data after dilation is mapped back to the actual input sensor data.

Algorithm 1 Component Labeling on Sparse Data

```
D: Input Lidar Data
D_b: Binary Lidar Data
D_d: Dilated Lidar Data
L_d: Labeled Data (after dilate)
L_r: Result Labeled Data

D ← load_lidar_data();
D_b ← binarize(D)
D_d ← dilate(D_b)
L_d ← labeling(D_d)
L_r ← map(D, L_d)
```

The pseudo-code of the proposed technique is presented in Algorithm 1. For labeling, a two-pass labeling algorithm [3] can be employed which is very simple to implement. A two-pass algorithm first stores temporary labels and equivalences. In the second pass, temporary labels are replaced by smallest equivalent label. But, the algorithm is not limited to it and other faster labeling algorithms [17] can also be employed.

IV. Application of Proposed Algorithm in Map Building

This section presents the application of the proposed algorithm in map building. In our previous work, [18], [19] we have proposed techniques to build maps from 2D Lidar data using clustering. Figure 5 shows a scan of a wall with an open door in between. In order to extract line features from the separated walls, clustering is applied on the Lidar data. Fuzzy c-means clustering [20] or FCM is employed to cluster Lidar data and extract the centroids. These centroids are joined to form a line. For distinct segments (such as walls), clustering can be applied on each segment and any number of clusters can be obtained. However, in case of open spaces in data, if the data is not segmented appropriately, the entire data-set can be mistaken as a single segment and inaccurate clusters will be formed. Figure 6 shows the result of applying fuzzy c-means clustering with 3 clusters. It can be seen that inaccurate number of clusters (i.e. 3 in this case) results in data clustering which omits the gaps present in the data (cluster indicated by blue color spans over the other segment omitting the gap in between as shown in Figure 6). The cluster centroids are shown in red dots in the figure. Line map obtained after joining the centroids is shown in Figure 7. Directly joining the centroids results in error in the map shown in Figure 7 where the gap between the line is not taken into consideration.

By applying the proposed algorithm, the Lidar data is first dilated and the result is shown in Figure 8. Next, labeling in applied and two labels are obtained as shown in Figure 9. In the next step, fuzzy c-means clustering is applied in each data segment. Since the input data has been segmented into two segments, any number of centroids can be obtained by applying clustering separately in each segment. Six and four number of clusters have been employed on labels 1 and 2, respectively, as shown in Figure 10. The centroids obtained after labeling are joined in each data segment to obtain the final map as shown in Figure 11.

A brief description of fuzzy c-means clustering [20] is presented. FCM is an extension of the k-means [21], [22] algorithm, in which every point in the data set belongs to every cluster with a certain belonging factor. This factor is 0 if the data point doesn't belong to the cluster, and 1 if it absolutely belongs to the cluster. In this way, each point in the data set belongs to each cluster by a certain degree. FCM tries to optimize the following objective function:

$$\mathrm{J_m} = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \ , \ 1 \le m \le \infty \tag{5}$$

where $m \in \mathbb{R}$ and $m \geq 1$, $u_{i,j}$ is the degree membership of x_i in the cluster j, x_i is the i^{th} element and c_j is the center of the cluster. The similarity of measured data and the center is expressed by $\|*\|$.

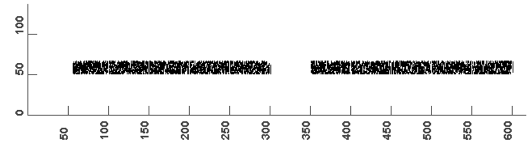


Fig. 5: Raw Lidar scan with gap.

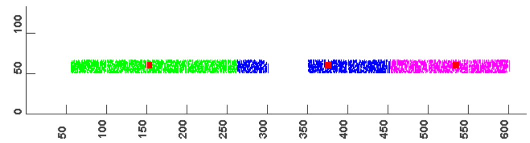


Fig. 6: Incorrect data clustering (3 clusters).

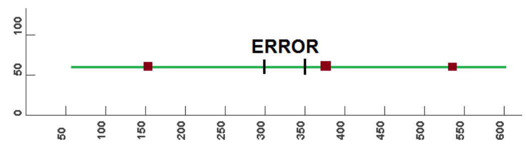


Fig. 7: Error while joining centroids.

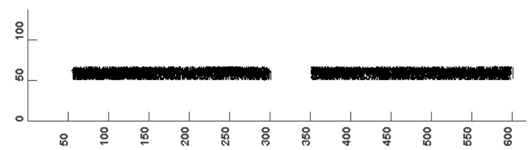


Fig. 8: Dilate on raw data.

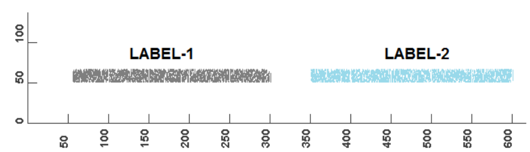


Fig. 9: Labeling on dilated data.

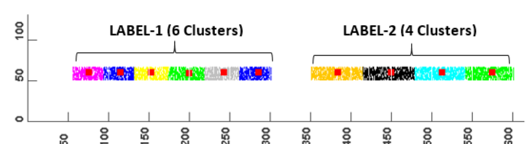


Fig. 10: Clustering on each label separately.

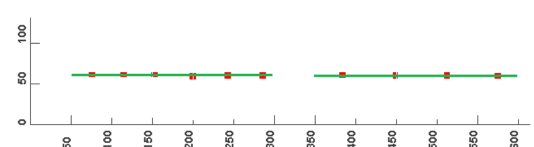


Fig. 11: Centroids joined in each label.

The objective function J_m is iteratively optimized by,

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad , \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}. \tag{6}$$

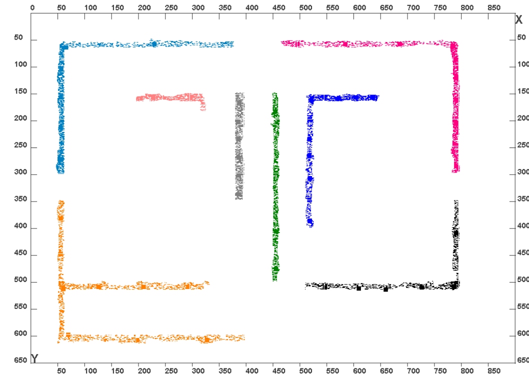
The iteration stops when the process converges to a local minimum J_m. The condition is reached when $\max_{ij}\left\{\left|u_{ij}^{k+1} - u_{ij}^{k}\right|\right\} < \varepsilon$ is satisfied, where the value of the terminating factor ε lies between 0 and 1.

V. Result

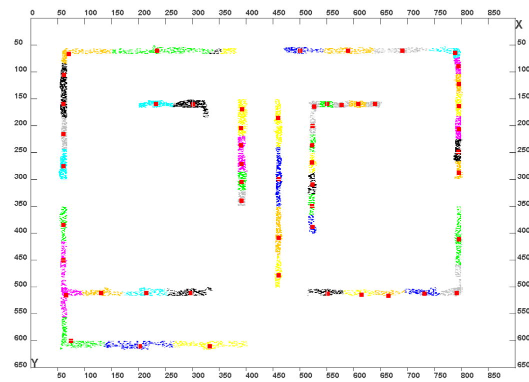
We extended the component labeling algorithm for sparse Lidar data using mathematical morphological operation of dilation. As shown in Figure 4, sparse Lidar data can be segmented using the proposed method. Generally dilate operation results in increasing the amount of input data, but we map the labels of dilated data with the original data, thus the algorithm does not alter the input data size. Moreover, dilate operation is not very computationally expensive, so it does not alter the time-complexity of component labeling. The core component of labeling can be replaced by faster versions with parallel implementations like [17]. Figure 12 shows the results of map building using the proposed algorithm. Figure 12(a) shows the result of raw Lidar data segmented by the proposed extended labeling algorithm. Result of clustering on the data segments is shown in Figure 12(b) and the final map is shown in Figure 12(c) with the centoids joined. Labeling of sparse Lidar data shown in Figure 12(a) using the proposed method was completed in 15 milliseconds. The proposed algorithm has a drawback that it takes extra memory equal to the size of the input data to store the dilated data.

VI. Conclusion

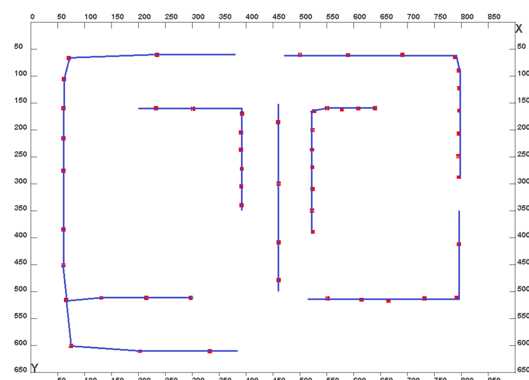
This paper proposed an extended component-labeling algorithm to label sparse Lidar data. The proposed algorithm requires a pre-processing step in which dilation is performed on the input Lidar data and then labeling is performed on the dilated data. Later, we showed the applicability of the proposed algorithm in building maps, where we showed that the algorithm can help detect gaps present in the line map due to open doors or narrow passages. Although the proposed algorithm requires more memory to store dilated data, improvements to remove this limitation and comparison with other techniques is considered as future work.



(a) Result of data segmented into 8 segments.



(b) Result of FCM clustering applied on segmented map.



(c) Resultant map obtained after joining centroids.

Fig. 12: Results of the proposed algorithm for map building.

References

[1] K. Ni and F. Dellaert, “Multi-level submap based slam using nested dissection,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 2558–2565.

[2] C. Fernández, V. Moreno, B. Curto, and J. A. Vicente, “Clustering and line detection in laser range measurements,” *Robot. Auton. Syst.*, vol. 58, pp. 720–726, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2009.10.008

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[4] M. B. Dillencourt, H. Samet, and M. Tamminen, “A general approach to connected-component labeling for arbitrary image representations,” *J. ACM*, vol. 39, no. 2, pp. 253–280, Apr. 1992. [Online]. Available: http://doi.acm.org/10.1145/128749.128750

[5] Y. Ioannou, B. Taati, R. Harrap, and M. A. Greenspan, “Difference of normals as a multi-scale operator in unorganized point clouds,” *CoRR*, vol. abs/1209.1759, 2012. [Online]. Available: http://arxiv.org/abs/1209.1759

[6] M. Thuy and F. Puente Len, “Non-linear, shape independent object tracking based on 2d lidar data,” in *IEEE Intelligent Vehicles Symposium*, Xi’an, China, 3-5 June 2009.

[7] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, Sept 2004.

[8] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *Image Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 53–69, Jan 2008.

[9] D. Doria and R. Radke, “Filling large holes in lidar data by inpainting

depth gradients," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, June 2012, pp. 65–72.

[10] R. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 4, pp. 532–550, July 1987.

[11] A. K. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[12] S. Fu, H. ying Liu, L. fang Gao, and Y. xian Gai, "Slam for mobile robots using laser range finder and monocular vision," in *Mechatronics and Machine Vision in Practice, 2007. M2VIP 2007. 14th International Conference on*, dec. 2007, pp. 91–96.

[13] URG-04LX Technical Specifications, "Urg-04lx technical specifications," *Website*, Nov, 2014. [Online]. Available: http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/URG-04LX_spec_en.pdf

[14] C. Ronse and P. A. Devijver, *Connected Components in Binary Images: The Detection Problem*. New York, NY, USA: John Wiley & Sons, Inc., 1984.

[15] L. Di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Image Analysis and Processing, 1999. Proceedings. International Conference on*, 1999, pp. 322–327.

[16] A. Jain and B. Yu, "Document representation and its application to page decomposition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 294–308, Mar 1998.

[17] J.-M. Park, C. G. Looney, and H.-C. Chen, "Fast connected component labeling algorithm using a divide and conquer technique." in *Computers and Their Applications*, S. Y. Shin, Ed. ISCA, pp. 373–376.

[18] A. Ravankar, Y. Hoshino, T. Emaru, and Y. Kobayashi, "Map building from laser range sensor information using mixed data clustering and singular value decomposition in noisy environment," in *System Integration (SII), 2011 IEEE/SICE International Symposium on*, dec. 2011, pp. 1232 –1238.

[19] A. A. Ravankar, Y. Hoshino, T. Emaru, and Y. Kobayashi, "Robot mapping using k-means clustering of laser range sensor data," *Bulletin of Networking, Computing, Systems, and Software*, vol. 1, no. 1, pp. pp1–9, 2012.

[20] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 23, pp. 191 – 203, 1984.

[21] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[22] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," 2000.