A Method of Generating Multi-Scale Disc-Like Distributions for NDT Registration Algorithm

Hyunki Hong and Beomhee Lee
Department of Electrical and Computer Engineering (ASRI), Seoul National University, Seoul, Republic of Korea
Email: {ckowl, bhlee}@snu.ac.kr

***Abstract*—Thick distributions whose mass centers are hanging in the air is one reason lowering the accuracy of Normal Distributions Transform (NDT). This paper proposes a method to solve this problem. The proposed method is transforming reference point cloud to disc-like distributions fitting the structure of point cloud. The subdivision of cells is based on octree, and it subdivides point cloud by different sizes of cells. By Principal Component Analysis (PCA), it finds fitting plane for the partial reference point cloud in each cell. It compares the Mean Square Error (MSE) to the threshold value τ to evaluate the flatness of the partial point cloud. If the cloud is not flat, the corresponding cell is subdivided to smaller cells, and those cells subdivide the cloud once more. The proposed method recursively subdivides reference point cloud until no more partial point cloud is possible to be subdivided. The experiment evaluates the loss of data and the error related to τ. In addition, the flat distributions transformed from reference point cloud is shown.**

***Index Terms*—point cloud, registration, scan matching, normal distributions transform, NDT, subdivision, multi-scale, disc-like, flat**

I. Introduction

Normal Distributions Transform (NDT) is one of the most famous registration algorithms [1]-[3]. The algorithm registers a point cloud to the reference point cloud. In robotics, it is a method of estimating the pose variation of a robot. While the robot continuously scans the environment, it registers each new point cloud to preceding point cloud or the global map by registration algorithm. The transformation vector is obtained by each registration, and it is the estimation of pose variation.

NDT is a registration algorithm accelerated by transforming reference point cloud to multiple distributions. Three processes of NDT and variants are subdividing the reference point cloud to partial point clouds, transforming partial point clouds into distributions, and registering new point cloud to distributions. At first, NDT subdivides reference point cloud by regular cells. Next, it computes means and covariance matrices of partial point clouds in cells, and it transforms them into normal distributions. Finally, it registers new point clouds by maximizing the sum of scores.

Manuscript received April 10, 2015; revised October 24, 2015.

By comparison with Iterative Closest Point (ICP), the most popular registration algorithm, the process of searching for correspondence in NDT is faster since the number of distributions is small [4]. However, the accuracy of NDT is lower than ICP since the group of distributions is an approximate representation of structure. In the registering process of NDT, the new point cloud gets closer iteratively by maximizing the score. Therefore, new points tend to move to the mass center of reference distribution where the score is the maximum.

The big issue of NDT is the size of cells. If the side length of regular cell is bigger, the structure of reference point cloud in a cell is more complex [2], [4]. Thus the mass center of the point cloud is possibly far away from the point cloud. For example, in Fig. 1, a point cloud in a cell is a wall in L-shape. The ball in Fig. 1(a) is an ellipsoid whose radius are Mahalanobis distance of 1.6. It is obvious that the mass center is hanging in the air, and that distribution is not fitting the point cloud. Therefore, new points tending to move to the mass center would lead to the failure of registration or the low accuracy. On the other hand, if the cells are subdivided once more as shown in Fig. 1(b), distributions in cells are disc-like, and mean points are almost near the centers of partial point clouds. Thus, distributions generated by the small cells represents the structure better than by the big cells. However, if the number of reference points in a cell is too small, the distribution hardly represent the structure, and it could even be generated.

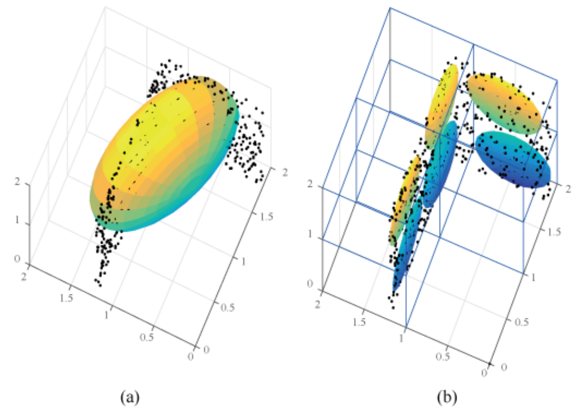


Figure 1. (a) A point cloud whose structure is not planar generates a thick distribution. (b) The desired flat disc-like distributions.

This paper focuses on the effect of cell size and propose a new method to generate multi-scale disc-like

doi: 10.18178/ijmerr.5.1.52-56

distributions for NDT. To decrease the number of distributions and represent the structure of environment better, the proposed method subdivides reference point cloud by multi-scale cells. A big plane-like partial point cloud would be represented by one disc-like distribution while the complex-shaped one would be represented by distributions in multiple sizes. The most important part of the proposed method is the flatness evaluation method. It determines whether the partial point clouds are flat.

The plan of the paper is as follows. In Section II the processes of the proposed method is described, and the details about octree-based subdivision and flatness evaluation method are described. In Section III the results of visualization is demonstrated, and the error and loss of data varied by threshold value τ is discussed.

II. Background

The process of NDT could be divided into subdividing, generating, and registering [5]. In the subdividing process, NDT generates cells and subdivides reference point cloud into partial point clouds by those cells as shown in Fig. 2(b). NDT transforms each partial point cloud into a normal distribution whose score function is

$$s(\mathbf{p}) = \exp\left(-\frac{1}{2}(\mathbf{p}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{p}-\mathbf{m})\right) \tag{1}$$

where $\mathbf{m}$ is the mean vector, and $\mathbf{C}$ is the covariance matrix. $\mathbf{m}$ and $\mathbf{C}$ are computed by

$$\mathbf{m} = \frac{1}{N_{ref}}\sum_{i=1}^{N_{ref}} \mathbf{p}_{ref,i} \tag{2}$$

$$\mathbf{C} = \frac{1}{n_{ref}-1}\sum_{i=1}^{n_{ref}} \left(\mathbf{p}_{ref,i}-\mathbf{m}\right)^2 \tag{3}$$

where $\mathbf{p}_{ref,i}$ is the ith reference point, and n_{ref} is the number of $\mathbf{p}_{ref}$ in the cell. In the generating process, NDT computes both $\mathbf{m}$ and $\mathbf{C}$ of each partial point cloud to generate distributions as shown in Fig. 2(c). In registering process, NDT computes the transformation vector ξ^* which maximizes the sum of scores

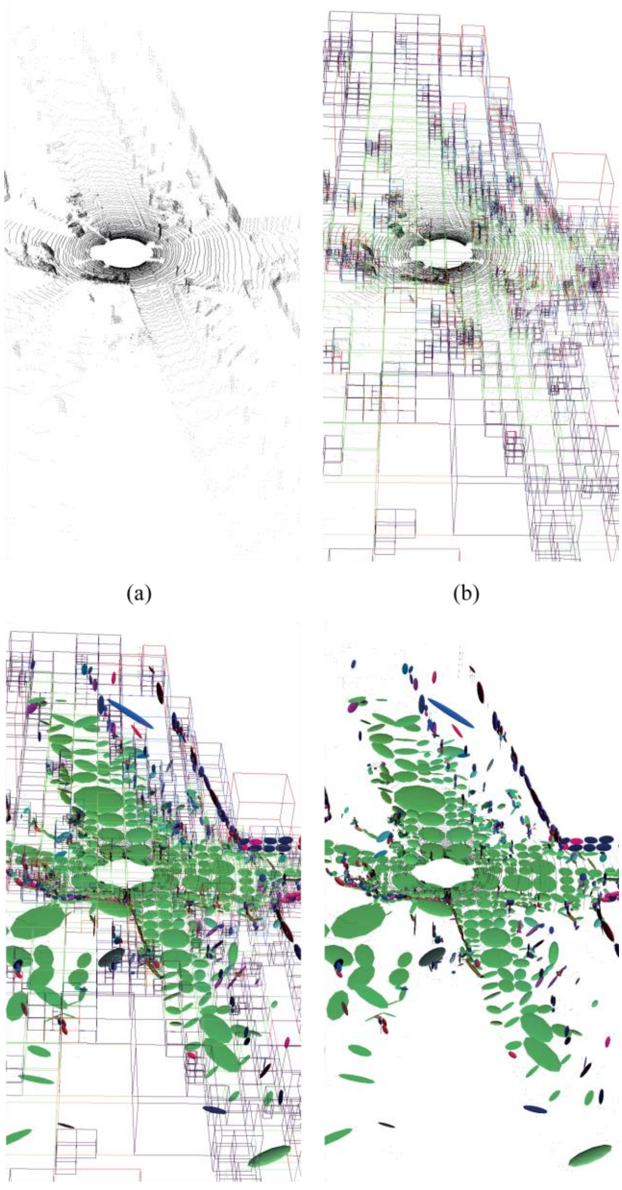
$$S = \sum_{i=1}^{N_{new}} s(T(\mathbf{p}_{new,i}, \xi)) \tag{4}$$

where $\mathbf{p}_{new,i}$ is the ith new point, N_{new} is the number of new points. $T(\mathbf{p},\xi)$ is the transformation function as

$$T(\mathbf{p},\xi) = R(\xi)\mathbf{p} + t(\xi) \tag{5}$$

where $R(\xi)$ is the rotational matrix, and $t(\xi)$ is the translational vector. To obtain ξ^*, Newton method and Levenberg-Marquardt algorithm are usually used [2], [5].

In some cases, the error between the estimated pose variation and ground truth is not acceptable. One of reasons is that NDT reaches a local maximum, and another reason is that the global maximum is wrong because of the bad map representation of normal distributions. This paper focus on the second reason and proposes a method to improve the map representation of distributions.



(a) (b) (c) (d)

Figure 2. (a) A reference point cloud. (b) Octree cells subdividing the reference point cloud. (c) Octree cells and distributions transformed from reference point cloud. (d) Disc-Like distributions. The color is RGB values mapped from normal vectors.

III. The Proposed Method

The goal of the proposed method is generating disc-like distributions to represent the structure of point cloud as shown in Fig. 2(d). It generates disc-like distributions by subdividing point clouds until all of partial point clouds are disc-like. To generate disc-like distributions, it finds the fitting planes for partial point clouds and evaluate their flatness. If a point cloud is flat, it would not be subdivided. On the other hand, if a point cloud is not flat, it would be subdivided into two or more partial point clouds by smaller cells. It would be recursively repeated until all partial point clouds are flat. The process is as follows:

1. Compute the side length of the initial cell including all reference points.
2. Evaluate the flatness of each partial point cloud in a cell.
3. Evaluate flatness of partial point clouds. If a partial point cloud is not flat, then subdivide the corresponding cell.
4. Subdivide the point clouds by subdivided cells.
5. Repeat 2-4 recursively. Terminate recursion if partial point clouds are flat or degenerated.

The evaluation at step 2 is based on the flatness evaluation method, and the cells and reference point cloud are subdivided by octree-based subdivision method at step 3 and 4. The details about two methods would be described in subsections.

A. *Flatness Evaluation Method*

The flatness evaluation of a point cloud is the important part of the proposed method. It decides whether the point cloud need be subdivided. If the shape of a partial point cloud is not flat, then it need be subdivided. The reference point cloud is recursively subdivided until all of partial point clouds are flat or degenerated.

The criteria of flatness evaluation is the Mean Square Error (MSE) of distances between reference points and the corresponding fitting plane. It is obtained as follows:

1. Compute $\mathbf{m}$ and $\mathbf{C}$.
2. Find the fitting plane.
3. Compute MSE and evaluate the flatness.

First of all, it computes mean and covariance of each partial point. Next, it computes the normal vector $\mathbf{n}$ of the fitting plane by principal component analysis (PCA) as shown in Fig. 3 [6]. It uses the eigenvector corresponding to the smallest eigenvalue of covariance matrix $\mathbf{C}$ as $\mathbf{n}$, and then it obtains a plane which includes $\mathbf{m}$ as follows:

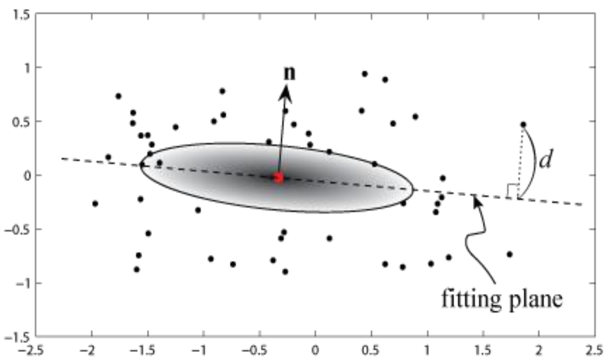


Figure 3. Flatness evaluation method computes MSE of distance between reference points and the fitting plane to evaluate the flatness of the partial point cloud in a cell. The black points are reference points, and the red point is the mean point.

$$\mathbf{n}^T\mathbf{p} = \mathbf{n}^T\mathbf{m} \tag{6}$$

where $\mathbf{p}$ is an arbitrary point on the plane. Finally, it computes the MSE of distances. The square of the distance d between a point and the plane as shown in Fig. 3 is obtained by

$$\begin{aligned} d^2 &= \left|\mathbf{n}^T(\mathbf{p}-\mathbf{p}_m)\right|^2 \\ &= \mathbf{n}^T(\mathbf{p}-\mathbf{p}_m)(\mathbf{p}-\mathbf{p}_m)^T\mathbf{n} \end{aligned} \tag{7}$$

MSE is the mean of d^2, and it is computed as follows [7].

$$\begin{aligned} MSE &= \frac{1}{N_{ref}}\sum_{i=1}^{N_{ref}} d^2 \\ &= \frac{1}{N_{ref}}\sum_{i=1}^{N_{ref}}\left(\mathbf{n}^T(\mathbf{p}-\mathbf{p}_m)(\mathbf{p}-\mathbf{p}_m)^T\mathbf{n}\right) \\ &= \frac{N_{ref}-1}{N_{ref}}\mathbf{n}^T\left(\frac{1}{N_{ref}-1}\sum_{i=1}^{N_{ref}}\left((\mathbf{p}-\mathbf{p}_m)(\mathbf{p}-\mathbf{p}_m)^T\right)\right)\mathbf{n} \\ &= \frac{N_{ref}-1}{N_{ref}}\mathbf{n}^T\mathbf{C}\mathbf{n} \end{aligned} \tag{8}$$

As shown in (8), in fact, it reuses $\mathbf{C}$ instead of computing each square of distance. Therefore, the elapsed time of flatness evaluation would be reduced. For the proposed method, if MSE is bigger than the threshold τ, it means that the big number of points are further than the threshold distance. In short, the variance of point cloud is too big; therefore, it is not disc-like.

The method has a trade-off between runtime and the accuracy. If τ is small, then the examination becomes strict. As a result, the runtime becomes longer since the reference point cloud is subdivided more times. Moreover, the part of point clouds would be degenerated since it transforms partial point clouds to distributions only if the number of points is bigger than 4.

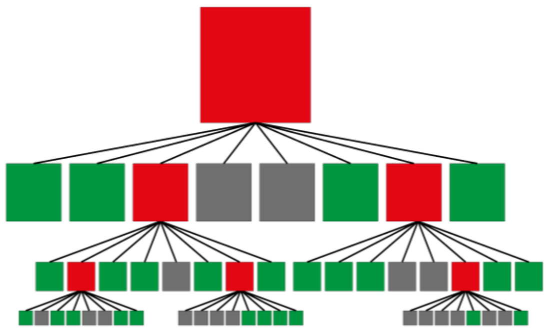


Figure 4. The octree structure. Red nodes are subdivided since the distributions are not disc-like. Gray nodes are degenerated since they have points less than 4. Green nodes are not subdivided since the distributions are flat.

B. *Octree-Based Subdivision Method*

Octree structure as shown in Fig. 4 is one of subdivision methods for NDT [2], [8]. It is effective to manage partial point clouds. In addition, it is efficient to search for desired cells and points. For those reasons, the proposed method uses octree to generate cells subdividing the reference point cloud. The first step of octree is setting the range. That is, it need compute the side length of initial cell including all of reference points. The top square in Fig. 4 stands for this initial cell. To balance the tree, it is a strategy setting the center at the average point of reference points. Each cell is split to 8 cells by xy, yz, and xz planes through the center of cell. And then, reference points in that cell are subdivided by those subdivided cells.

In Fig. 4, squares in red mean that the partial point clouds should be subdivided since their structures are not flat. The green squares are stand for flat point clouds, and the gray ones are degenerated point clouds, whose number of points are less than 4.

IV. Experiments

The proposed method is implemented based on Point Cloud Library (PCL) [9]. The benchmark data set is the scenario 1 provided by Karlsruhe Institute of Technology, and it is scanned outdoor by Velodyne range sensor [10]. 100 scans are randomly chosen for the experiment. In this experiment, the average percentage of loss of points and the average errors of fitting planes related to threshold τ are evaluated.

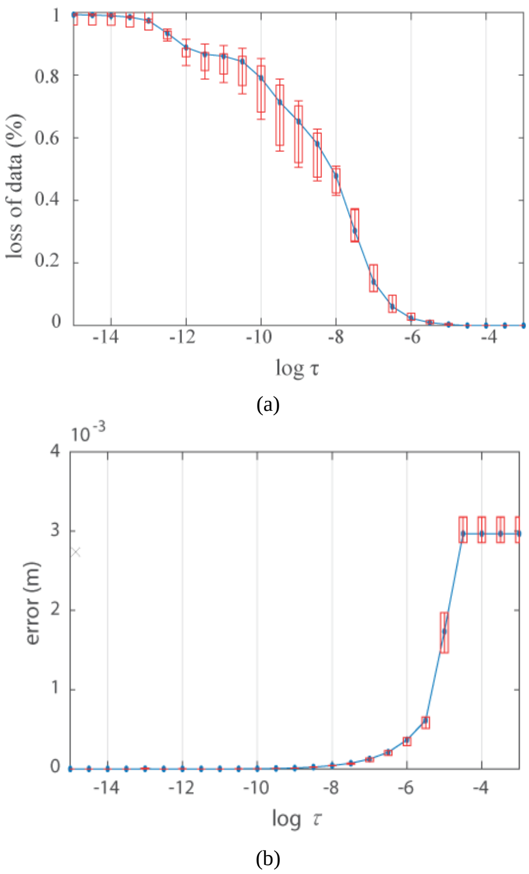


Figure 5. (a) $\log\tau$ - loss of data. (b) $\log\tau$ - error.

The results are as shown in Fig. 5. The error decreased sharply as $\log\tau$ becomes small. When $\log\tau$ is smaller than -6, the error is acceptably small. However, as $\log\tau$ decrease, the loss of data increases. The reason is that more partial point clouds are degenerated since the flatness evaluation becomes strict as the threshold value decreases. Therefore, $\log\tau$ is better smaller than -7. In the interval from -12 to -8, the variation of loss of data becomes large. Cases whose loss of data increases steeply as $\log\tau$ decreases are scanned at complex places. By the experiment, the partial point clouds scanned at more complex environments are much easier to be degenerated by the strict threshold value, and it leads to the big loss of data. Fig. 2 is an example when τ is set to 10^{-6}. The radius of ellipsoids standing for distributions are Mahalanobis distance of 1.8. The different sizes of distributions are all disc-like. The RGB color of distributions are related to the normal vectors of fitting plane.

V. Conclusion

This paper proposes a method to generate multi-scale disc-like distributions for NDT. It uses octree algorithm to easily subdivide reference point cloud. In addition, it uses PCA to find the normal vector for a fitting plane. MSE is used to evaluate the flatness of point clouds. If a partial point cloud is not flat, then it subdivides the point cloud recursively. As the results shown in the experiment, the loss of data steeply increases as the threshold value τ is decreased. In addition, the error increases as the threshold value τ is increased.

Acknowledgment

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2013R1A2A1A0500 5547), in part by ASRI, and in part by the Brain Korea 21 Plus Project.

References

[1] P. Biber and W. Straßer, “The normal distributions the normal distributions,” in *Proc. IEEE/RSJ Int. Conf. on Intell. Robot. and Syst.*, Las Vegas, 2003, pp. 2743-2748.

[2] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3d-ndt,” *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[3] E. Takeuchi and T. Tsubouchi, “A 3-d scan matching using improved 3-D normal distributions transform for mobile robotic mapping,” in *Proc. IEEE/RSJ Int. Conf. on Intell. Robot. and Syst.*, Beijing, 2006, pp. 3068-3073.

[4] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, “Evaluation of 3d registration reliability and speed-a comparison of icp and ndt,” in *Proc. IEEE/RSJ IEEE International Conference on Robotics and Automation*, 2009, pp. 2263-2268.

[5] *Int. Conf. on Intell. Robot. and Syst.*, Kobe, 2009, pp. 3907-3912

[6] C. Ulaş and T. Hakan, “3D multi-layered normal distribution transform for fast and long range scan matching,” *J. Intell. & Robot. Sys.*, vol. 71, no. 1, pp. 85-108, 2013.

[7] R. O. Duda and E. H. Peter, *Pattern Classification and Scene Analysis*, 1st ed. New York: Wiley, 1973.

[8] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, “Fast plane detection and polygonalization in noisy 3D range images,” in *Proc. IEEE/RSJ Int. Conf. on Intell. Robot. and Syst.*, Nice, 2008, pp. 3378-3383.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189-206, 2013.

[10] A. Aldoma, Z. C. Marton, F. Tombari, W. Wohlkinger, C. Pothast, B. Zeisl, *et al.*, "Point cloud library," *IEEE Robotics & Automation Magazine*, 2012.
[11] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 393-398.

Hyun K. Hong received his B.S. in Electrical and Computer Engineering from Seoul National University in 2012. He is currently a M.S candidate in the Department of Electrical and Computer Engineering at Seoul National University. His current research interests include graph SLAM and multi-agent control.

Boem H. Lee received the B.S. degree and M.S. degree in electronics engineering from Seoul National University, Seoul, the Republic of Korea in 1978 and 1980, respectively, and the Ph.D. degree in computer, information, and control engineering from the University of Michigan, Ann Arbor, MI in 1985.

He was an Assistant Professor with the School of Electrical Engineering at Purdue University, West Lafayette, IN from 1985 to 1987. He joined Seoul National University in 1987. He is currently a Professor with the Department of Electrical and Computer Engineering. His research interests include multi-agent system coordination, control, and application.

Prof. Lee has been a Fellow of the Robotics and Automation Society since 2004.