Where Am I: Localization and 3D Maps for Autonomous Vehicles

Farzeen Munir, Shoaib Azam, Ahmad Muqeem Sheri, YeongMin Ko and Moongu Jeon
School of Electrical Engineering and Computer Science,
Gwangju Institute of Science and Technology, Republic of Korea

Keywords: Localization, 3D Point Cloud, Maps, NDT Matching, Autonomous Vehicles.

Abstract: The nuts and bolts of autonomous driving find its root in devising the localization strategy. Lidar as one of the newest technologies developed in the recent years, provides rich information about the environment in the form of point cloud data which can be used for localization. In this paper, we discuss a localization approach which generates a 3D map from Lidar's point cloud data using Normal Distribution Transform (NDT) mapping. We use our own dataset collected using our self driving car KIA Soul EV equipped with Lidar and cameras. Once the 3D map has been generated, we have used NDT matching for localizing the self driving car.

1 INTRODUCTION

Autonomous driving is one of the hot research topics captivating the focus of most leading companies and researchers, and also is a comprehensive research venture involving interdisciplinary study. The four fundamental research questions in autonomous driving are: where am I? What are around me? What will happen next? and what should I do?. The quest to find the solution of each one of them is a step towards successful autonomous driving.

The first and most fundamental question "where am I? " signifies the importance of localization and mapping for autonomous driving (Jo et al., 2015). For the car to drive without human intervention, it needs to locate itself in the 3D environment so that it could navigate. Autonomous vehicles need to come a long way to have the capability to make real-time decisions. In order to achieve it, mapping and localization become critical components of autonomous driving (Karlsson and Gustafsson, 2017).

Humans require only simple 2D map for navigation. This 2D map information is not enough for autonomous vehicles to navigate in the environment. This hindrance of not enough information is compensated with the availability of high definition 3D maps that include information about road boundaries, lanes, building, curb height. Besides that, this information of environment needs to be precise and accurate. The autonomous vehicles are equipped with Lidar and camera sensors that provide the information about the surroundings. Lidar provides accurate 3D point cloud of the environment which is used to make an offline 3D maps (Munir et al., 2017).

In literature, three techniques are popular to make 3D maps from point cloud data. It includes Iterative Closet Point matching (ICP) (Chetverikov et al., 2002), Normal Distribution Transform matching (NDT) (Ulaş and Temeltaş, 2013) and Simultaneous Localization and Mapping (SLAM) (Dissanayake et al., 2001).

(Borrmann et al., 2008) introduced SLAM algorithm to map the environment, by detecting the pair of points in the two point-clouds and minimized the distance. The loop detection was also incorporated into the algorithm. (Kim et al., 2018) used 2D hector SLAM technique to map the environment in real time. The mobile robot was equipped with hybrid laser scanning system to model the 3D map.

The base algorithm for ICP was proposed by (Besl and McKay, 1992). It used 6 degree of freedom to find the closest point to the geometric entity from a given point. (He et al., 2017) proposed a modified version of ICP to register 3D scan point cloud which used geometric features of the point cloud to register. Since the generic ICP algorithm requires good initialization and approximate registration, it is hard to estimate. The use of geometric features in modified ICP improve the accuracy as compared to generic ICP.

(Biber and Straßer, 2003) introduced NDT for matching 2D laser scan. It transformed the points in the laser scan into a piecewise continuous and differentiable probability density. The probability density contains a set of the normal distributions. To match two scans, NDT sum is maximized. Many modified versions of NDT have been proposed by changing the minimization function. (Prieto et al., 2017) used NDT with the differential evolution to minimize the error

Munir, F., Azam, S., Sheri, A., Ko, Y. and Jeon, M.
Where Am I: Localization and 3D Maps for Autonomous Vehicles.
DOI: 10.5220/0007718404520457
In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, pages 452-457
ISBN: 978-989-758-374-2

between NDTs of two-point clouds. (Zhou et al., 2018) published a modern 3D library for processing point cloud.

In this paper, we have used VLP-32 Lidar to collect 3D point cloud data to make a map of Gwangju Institute of Science and Technology (GIST). The NDT transform is used to register two point clouds. The 3D map is generated to provide the reference to the localization algorithm used in the self-driving car. The overall framework for 3D map generation and localization is shown in Figure 1.

The rest of the paper is organized as follows. Section II describes NDT transform. Section III gives details of the experimental setup and data collection. Section IV explains experimental results, and Section V concludes the paper.

2 NORMAL DISTRIBUTION TRANSFORM (NDT)

3D point cloud matching is an integral part of map generation and localization. 3D maps enable self-driving cars to localize themselves in the environment. NDT mapping of consecutive scans is the most efficient way to make 3D maps. The brief explanation of normal distribution transform is given below, and more details are given in (Takeuchi and Tsubouchi, 2006).

The input point cloud denotes a new point cloud and reference point cloud denotes already built map using past point clouds.

2.1 Normal Distribution Transform

NDT assigns each point in point cloud to a voxel. A voxel is a 3D lattice cube to which points are assigned depending upon their coordinate value. The Point cloud is divided into k ND voxels. M_k is the number of points in a voxel k and $\mathbf{x}_i = (x_i, y_i, z_i)^t (i = 0...M-1)$ is a coordinate vector for each point in the ND voxel. Equation 1 and 2 give $\mathbf{p}_k$, the average coordinate of ND voxel k and $\sum_k$, the covariance of ND voxel k. The estimation value $e(\mathbf{x})$ of a point in the ND voxel k is given by equation 3.

$$\mathbf{p}_k = \frac{1}{M_k} \sum_{i=0}^{M_k-1} \mathbf{x}_{k_i} \tag{1}$$

$$\sum\nolimits_k = \frac{1}{M_k} \sum_{i=0}^{M_k-1} (\mathbf{x}_{k_i} - \mathbf{p}_k)(\mathbf{x}_{k_i} - \mathbf{p}_k)^t \tag{2}$$

$$e(\mathbf{x}) := exp(-\frac{(\mathbf{x}-\mathbf{p}_k)^t \sum_k M_k (\mathbf{x}-\mathbf{p}_k)}{2}) \tag{3}$$

Algorithm 1: For mapping and matching 3D scan data.

Result: Aligned Point Cloud P_a
P_i:Input point cloud data
P_r:Reference point cloud data
I_p:Initial Position from Odometry
Initialization:
  Initialization of P_i using I_p
Allocating the structure V
Voxelization:
foreach *points* $p_i \in P_i$ **do**
  find the voxel $v_i \in$ V that contains p_i
  store p_i in v_i
end
foreach *voxel* $v_i \in V$ **do**
  Averaging Using Equation 1. Covariance Using Equation 2. Estimating using Equation 3.
end
Incremental Update:
foreach *voxel* $v_i \in V$ **do**
  if P_i *is aligned with* P_r **then**
    Incremental Update Using Equation 4 and 5.
  end
end
Registration:
while *not converged* **do**
  foreach *points* $p_i \in P_i$ **do**
    find the R and t
    evaluating the score using Equation 7.
    find P_a using Equation 6
    if P_a *converged* **then**
      return P_a
    else
      Update parameters (R,T)
    end
  end
end

2.1.1 Incremental Update of ND Voxel

The number of points in the reference point cloud increases as more input point clouds are combined. The computation of NDT becomes expensive and slow. To avoid this problem, incremental update of the ND voxel is performed using equation 4 and 5.

$$m_k = m_{old} + \mathbf{x}_{k_i}, S_k = S_{old} + \mathbf{x}_{k_i}\mathbf{x}_{k_i}^t \tag{4}$$

$$\mathbf{p}_k = \frac{m_k}{M_k}, \sum\nolimits_k = S_k - \frac{\mathbf{p}_k m_k^t}{M_k} \tag{5}$$

The incremental update parameters m_k and S_k represent the mean and covariance of the current scan

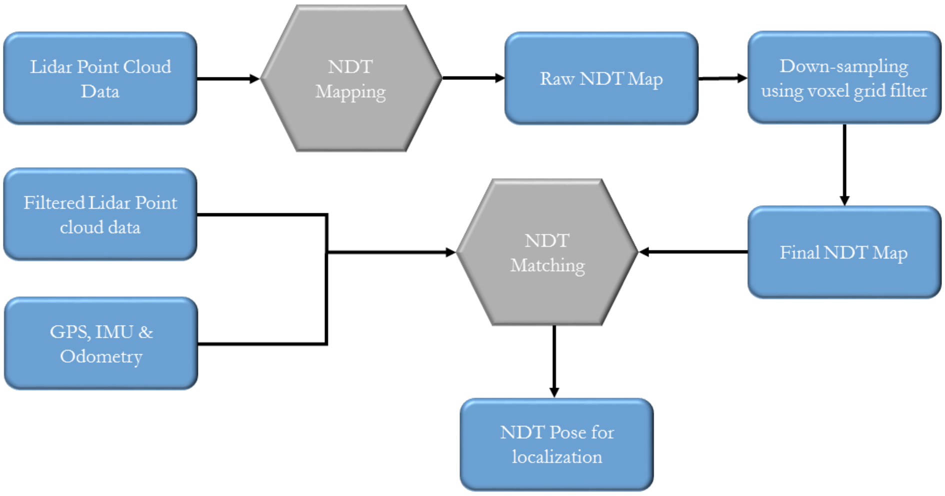


Figure 1: Overall framework for 3D map generation and localization. It takes raw point cloud as input and generates the 3D map using NDT mapping. It uses NDT matching to localize the self driving car using the information of 3D map and filtered point cloud data at the current time and searches for the best possible match between map and point cloud data.

Figure 2: MLV-Self Driving Car KIA Soul EV equipped with different sensors being used for collecting the data.

respectively. The m_k and S_k are maintained for each ND voxel k. They are updated when input point cloud is associated with the reference point cloud. The 3D coordinate equation to transform input point cloud is given by equation 6.

$$\mathbf{w}_i = R\mathbf{x}_i + t, \tag{6}$$

where R gives the rotation matrix to rotate euler angle α, β, γ along z, y, z axis. R is calculated as follows:

$$R = \begin{pmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{pmatrix} * \begin{pmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

and $t = (t_x, t_y, t_z)^t$ is the translation vector. The Euler angles and translation vector form the parameters of NDT mapping and given by vector $T = (\alpha, \beta, \gamma, t_x, t_y, t_z)^t$.

2.2 NDT Mapping and Matching

NDT mapping is the module of map generation. Its schematics is shown in Figure.3 , where the point clouds are converted to voxel using NDT and are combined together , and also the voxel grid filter is used to decrease the computation cost and to reduce the noise from the 3D map. More details of NDT mapping is described below.

1. Compute NDT of the reference point cloud.
2. Use initial position from odometry to set the initial position of input point cloud.
3. Apply NDT to input point cloud.
4. For the input cloud select corresponding ND voxel.

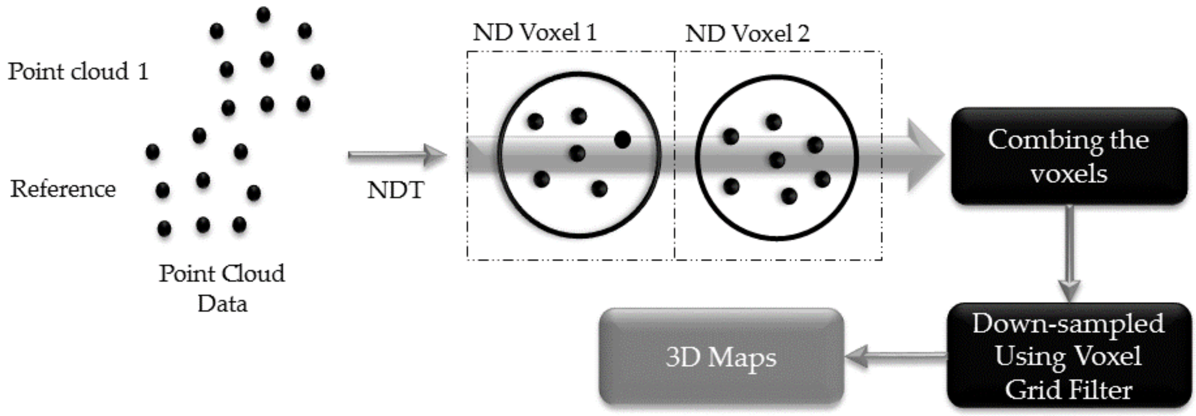


Figure 3: Procedure to make 3D maps for localization.

5. Use the Newton's method to update parameters.
6. If parameter converges go to 7, else go to 3.
7. Combine the input cloud with the reference point cloud.
8. Start again from step 2.

Now that we have a map and want to localize, we need to match our current location information to the generated map. The matching of the point cloud is a search problem, which finds the parameter $T=(\alpha,\beta,\gamma,t_x,t_y,t_z)$ that best transforms the input point cloud to the reference point cloud. An evaluation function given by equation 7 is used to evaluate the fitness of the input point cloud to the reference point cloud. Newton non-linear optimizer is applied to *E(X,T)*.

$$E(X,T) = \sum_{i=1}^{N-1} exp(-\frac{(\mathbf{w}_i - p_i)^t \sum_i^{-1}(\mathbf{w}_i - p_i)}{2}) \tag{8}$$

Algorithm 1 explains the NDT mapping and matching process.

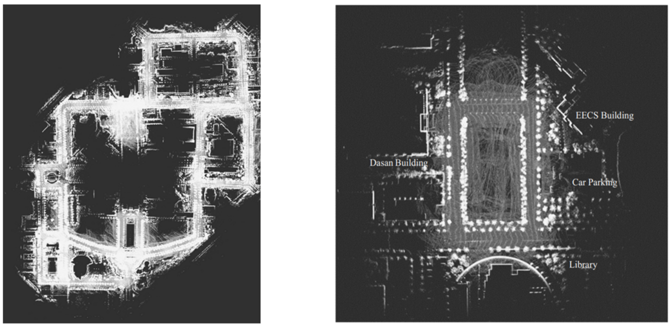
3 DATA COLLECTION

We have developed a platform for autonomous driving (Munir et al., 2018). It serves as the experimental setup for the collection of data, testing and evaluation of algorithms. KIA Soul EV is equipped with state-of-art equipment consisting of 32 channel Lidar, Novatel Global Navigation satellite System (GNSS) module and Cameras (Azam et al., 2017). VLP-32 Lidar has been used for collecting point cloud data of the environment (Azam et al., 2018). It has 32 channel lasers and detectors pairs. It rotates at frequency of 15hz and outputs data at 100Mbps. It can detect up to 100m. 300,000 points/sec are generated which specify the X,Y and Z coordinates of object in the surroundings in Cartesian coordinate system. The distance of each point is specified from center of Lidar. The data is collected of Gwangju Institute of Science and Technology (GIST) using MLV self driving car shown in Figure 2. Figure 6 shows the map of GIST, and red path specify the route where the car was driven to collect the data manually.

4 EXPERIMENTATION

Localization of self driving car in its environment is of immense importance. The use of point cloud data in generating the 3D maps solves this predicament of localization. Normal Distribution Transform mapping provides an efficient way to develop maps for the environment. For map generation, approximately 27GB of raw point cloud data was collected using Lidar. The generation of map for the whole area in a single attempt, requires a lot of computational power and time. NDT matching on a system having 12 cores CPU takes more than 24hrs to process less than 1GB data. The processing time has been greatly optimized by the use of GPU with the limitation that large memory is required for storing the map at runtime. An efficient method to solve memory issue is to make sub-maps and combine them later into a complete map. Each sub-map is downsampled to reduce the data points. The range accuracy of generated map depends on the point cloud data. Since the Lidar has 3cm range accuracy, the generated map is accurate upto 3cm. Figure 4 shows the generated map of GIST using the point cloud data.

For the localization, we have used the NDT matching that gives the estimated position and orientation of the vehicle by matching the best possible in-



(a) (b)

Figure 4: 3D map generated using NDT mapping.(a) Complete GIST map (b) Small portion of map generated for testing the autonomous driving.

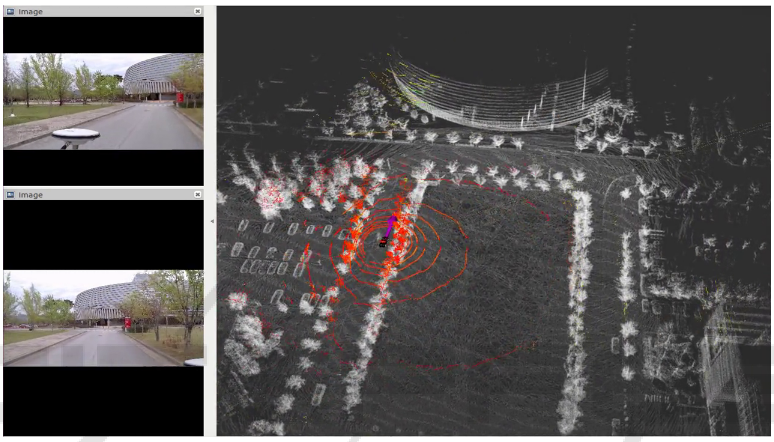


Figure 5: Localization using NDT matching. The arrow shows the pose of vehicle.

Figure 6: GIST Map showing the route of the self driving car.

formation between the scan data and the 3D map. Figure 5 shows the localization results of NDT matching.

5 CONCLUSION

The autonomous driving is the new future of artificial intelligence and many stakeholders are investing in this uprising research area. In this paper, we have discussed the localization for autonomous driving. For localization, we have to make the 3D map and depending on that map the self driving car can localize itself. We have used NDT mapping algorithm for generating the 3D map of GIST using our own self-driving car. Once the map is being built, we have used NDT matching as scanning algorithm for matching the current Lidar point cloud to the 3D map which is being built previously.

The future work includes the use of motion planning algorithm for path finding depending on the localization information. Incorporating detection results to the localization is also in the future work.

ACKNOWLEDGMENT

This work was supported by GIST Research Institute (GRI) grant funded by the GIST in 2019, and by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2014-0-00077, Development of global multitarget tracking and event prediction techniques based on realtime large-scale video analysis)

REFERENCES

Azam, S., Munir, F., Rafique, A., and Jeon, M. (2017). Multi-sensor data collection and data fusion: A Step towards Self Driving Car. In *The International Conference on Big data, IoT, and Cloud Computing*, Jeju, South Korea.

Azam, S., Munir, F., Rafique, A., Ko, Y., Sheri, A. M., and Jeon, M. (2018). Object modeling from 3d point cloud data for self-driving vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 409–414. IEEE.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

Biber, P. and Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. In *IROS*, volume 3, pages 2743–2748.

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., and Hertzberg, J. (2008). Globally consistent 3d mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142.

Chetverikov, D., Svirko, D., Stepanov, D., and Krsek, P. (2002). The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE.

Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241.

He, Y., Liang, B., Yang, J., Li, S., and He, J. (2017). An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features. *Sensors*, 17(8):1862.

Jo, K., Jo, Y., Suhr, J. K., Jung, H. G., and Sunwoo, M. (2015). Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3377–3392.

Karlsson, R. and Gustafsson, F. (2017). The future of automotive localization algorithms: Available, reliable, and scalable localization: Anywhere and anytime. *IEEE signal processing magazine*, 34(2):60–69.

Kim, P., Chen, J., and Cho, Y. K. (2018). Slam-driven robotic mapping and registration of 3d point clouds. *Automation in Construction*, 89:38–48.

Munir, F., Azam, S., Hussain, M. I., Sheri, A. M., and Jeon, M. (2018). Autonomous vehicle: The architecture aspect of self driving car. In *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing*, SSIP 2018, pages 1–5, New York, NY, USA. ACM.

Munir, F., Azam, S., Rafique, A., and Jeon, M. (2017). Automated Labelling of 3d Point Cloud Data. *Korean Information Science Society*, pages 769–771.

Prieto, P. G., Martín, F., Moreno, L., and Carballeira, J. (2017). Dendt: 3d-ndt scan matching with differential evolution. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 719–724. IEEE.

Takeuchi, E. and Tsubouchi, T. (2006). A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping. In *IROS*, pages 3068–3073. IEEE.

Ulaş, C. and Temeltaş, H. (2013). 3d multi-layered normal distribution transform for fast and long range scan matching. *Journal of Intelligent & Robotic Systems*, 71(1):85–108.

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*.