

Normal Distribution Transform with Point Projection for 3D Point Cloud Registration

Mouna Attia
Université de Tunis El Manar
Faculté des Sciences de Tunis
Labotory: URAPOP
2092, Tunis, Tunisie
Email: mounaattia.info@gmail.com

Yosr Slama
Université de Tunis El Manar
Faculté des Sciences de Tunis
Labotory: URAPOP
2092, Tunis, Tunisie
Email: yosr.slama@gmail.com

Abstract—Normal Distribution Transform (NDT) is a registration algorithm very used in many fields. Its main idea consists in modeling clouds (points sets) with a set of Gaussian distributions generated by discretizing each points set into voxels. This algorithm gives a good transformation estimation if a point and its neighbors represent a planar and linear patch. In light of this fact, this paper presents a new way to apply NDT on 3D registration problem. Our approach consists in applying NDT onto projected clouds instead of the original ones. The new point clouds are determined by taking into consideration the shape topology. The proposed approach uses a statistical method to find the best plane that represents each point cloud. An experimental study ,carried out on several clouds, shows that the proposed approach is better than the original NDT in terms of accuracy and runtime performance.

Index Terms—3D registration, Normal Distribution Transform (3D), point clouds, PCA, plane form.

I. INTRODUCTION

3D point cloud registration is a crucial building block in several applications solving many complex tasks. These tasks include Simultaneous Localization and Mapping (SLAM), object detection, control manufacturing, medical imaging, archaeology and recognition problem. Most of these applications require accurate results in short time. 3D point cloud registration can be formulated as the problem of finding the transformation that optimally positions two 3D point clouds representing the reference and the model, in a common coordinate system. We denote the model data by $U = \{u_i | i = 1..n\}$ where $u_i \in \mathbb{R}^3$ and the reference data by $V = \{v_j | j = 1..m\}$ where $v_j \in \mathbb{R}^3$. The registration goal is to determine the transformation denoted $Trans$ which is a $4 * 4$ matrix such that the Euclidian distance ED between U and $Trans(V)$ is as low as possible.

Several researches have treated this problem and proposed many algorithms. The most popular and used algorithms are ICP, NDT, Ransac and Softassign. Several writes studied and compared these algorithms. We cite in particular ATTIA et al. [1], which concluded that Iterative Closest Point (ICP) [2] and Normal Distribution Transform (NDT) [3] are the most suitable for the most of applications.

The two methods use iterative refinement techniques and involve three steps but each algorithm adopts its own strategy

for computing the transformation. In fact, ICP attempts to find transformation parameters in a way that the Euclidean distance between nearest neighbor points is minimized. This method is based of search of neighbors. This step is a bottleneck for ICP.

NDT algorithm represents the underlying point clouds as a set of Gaussian distributions that locally models the surface. It is a promising method which overcomes the bottleneck of ICP since it doesnt use the search of nearest neighbor. However, the accuracy of NDT depends strongly on the size of voxel (small cube containing a subset of points) used in the algorithm. In fact, if we have a little size of voxels, the clouds constitute a plane surface and thereafter the surface is perfectly modeled. NDT has also many pitfalls. In fact, to detect the point clouds, we use a 3D scanner which sometimes generates a million of points in short time per scan. The processing of these points needs a huge execution time. Another problem is the outliers and the noise which can affect the accuracy.

To address the shortcomings of these algorithms, several papers propose many solutions based on geometrical and statistical methods. Other improvements operate on point cloud data such as Scale Invariant Feature Transform (SIFT) [4], Speeded-up Robust Features (SURF) [5]. In this context, we propose to operate on point cloud but without using the feature.

The contribution of this work is to use a point clouds for plane shape. So, we need a method allowing to determine new point clouds that form a plane without losing the original shape. One of the simple and efficient method we can use, is the projection of point clouds onto plane. After that, the next issue is how we can define a plane that preserves the shape topology. Here, we decide to use the Principal Component Analysis (PCA) to define a good plane.

The remainder of this paper is organized as follows. In Section II, we give an overview of the state-of-the-art of NDT algorithm improvements . In Section III, we discuss the background of our research. In Section IV, we present the proposed approach algorithm. The experimental results are detailed and analyzed in Section V. Finally, we summarize our results and explore venues of future research.

II. NORMAL DISTRIBUTION TRANSFORM (NDT)

NDT algorithm is a promising new approach to perform 3D point cloud registration. It was introduced for the first time by Biber and Strasser [6] for 2D scan registration and mapping. It was later expanded by Magnusson [3] for 3D point clouds. The main idea is to represent the point cloud as a set of normal distributions. This is achieved by dividing the surface constituted by the cloud into regular cells and then computing both the average vector and covariance matrix of each cell. This procedure is also known as discretization or splitting step of the algorithm. In fact, the cloud environment (a voxel in 3D) is subdivided into cells, to each of which is assigned a normal distribution that locally models the cloud intensity. The result of this transformation is a piecewise continuous and differentiable probability density which can be used to align another scan using Newtons algorithm [7]. An important feature of NDT is that it avoids computing explicit correspondences; it simply matches a point to the cell in which it lies. NDT has three steps:

- **Initialization:**

This step consists in initializing the voxel size. The environment of the reference cloud is divided into voxels. To each voxel is assigned some points from there a normal distribution ,that locally models the probability of measuring a point, is computed in the next step. Then a covariance matrix and an average are computed for each voxel.

- **Matching Step:**

Using space subdivision, we generate a Gaussian distribution function of each voxel which describes the clouds dispersion. This function does not model the pieces accurately. The distributions only locally model the points within each voxel and may not capture broader features present within the scan. In this step,for each point in the model clouds, we search the voxel that verifies this distribution.

- **Transformation determination:**

To determine the transformation, we have to define the objective (cost) function. For NDT, the objective function is based on cloud intensity and the best estimation corresponds to the maximum intensity value of the image. For this reason, the transformation (combining translation and rotation), denoted $Trans$, is considered optimal if the sum of the normal distributions for all points (see below) with parameters 1 is maximum. This sum is called transformation score. It is defined as follows :

$$Score(Trans) = \sum_i \exp\left(\frac{-(u'_i - q_i) \Sigma_i^{-1} (u'_i - q_i)}{2}\right) \quad (1)$$

q_i : The average of the corresponding voxel.

Σ_i : The covariance matrix of the normal distribution

corresponding to point u'_i after applying NDT on the first scan.

u'_i : The point u_i mapped into the coordinate frame of the reference cloud (point cloud of V) according to the transformation $Trans$.

Finally, a nonlinear optimization is performed to determine the transformation parameters.

III. RELATED WORKS

NDT is a 3D registration algorithm that solves several problems posed by ICP. However, it suffers from some pitfalls. The first problem is the use of a regular grid which causes several fundamental problems such as the discontinuity of the score function (when a data scan point passes one of its cell boundaries, the value of the score function jumps). Das et al. [8] suggest an alternative method that modifies the score function to become continuous. In fact, they propose to employ greedy clustering for cloud partitioning thereafter that there are few distributions. Thus, the modified score function includes the scores of all of the normal distributions for each point in the cloud.

Another improvement was introduced by Staylov et al. [9]. It is the distribution-to-distribution registration approach which transforms the reference cloud as well as the model cloud into normal distributions.

The second fundamental problem posed by NDT is that the registration performance relies on the cell size of the regular grid. Ulas et al. [10] propose a solution based on multi layered grid (ML-NDT). In fact, they suggest changing the cell size from large to small during the iterative optimization process.

A more serious problem, though, is that a normal distribution in each voxel does not represent the local structure of the point subset within the voxel accurately because the regular grid does not take into account the structures of the model scan. When part of the cloud data, i.e., that composed of surfaces, is modelled by a normal distribution, the shape of the point subset which minimizes information loss is a plane. Kim et al. [11] propose the supervoxel-NDT (SV-NDT) to solve the problem in which the normal distributions generated by a regular grid do not represent the local surface structures of the model cloud accurately. SV-NDT exploits the 3-D supervoxel segmentation technique to use local surface structures to partition the model cloud. In addition, the criterion of matching each point in the reference cloud to the corresponding distribution is modified using the local geometries of the reference cloud.

IV. PROPOSED METHOD

A. Motivation

Referenced to Kim et al. [11], in order to use the merits of the NDT and minimize the loss of information, the model cloud should be divided into locally planar surfaces. To have planar surfaces, many searches proposed to minimize the voxel size. Using a little voxel, we have two problems: (i) The execution time may be huge because of the enormous number of voxels and (ii) some voxels can have only one or

two points with environmental clouds. For these reasons, we propose to transform both the model and reference clouds onto planar surfaces without minimizing the voxel size. In order to transform the original clouds to planar clouds, we propose to project cloud onto plane preserving the shape topology.

B. Method Description

The proposed method follows three steps which are : defining projection plane, projection clouds onto plane and applying NDT onto projected clouds

- **Define projection plane:**

Similarly to our previous work presented in [12], we propose here to use the principal Analysis Component(PCA) method to define a projection plane. PCA [?] is a mathematical tool from applied linear algebra. It is an analysis method of multivariate statistical which can reduce the space dimension. It is based on the correlation among variables. It is often used in classification and compression techniques to project data on a new orthonormal basis in the Direction of the largest variance (DLV) [13]. The DLV corresponds to the largest eigenvector of the covariance matrix of the data, whereas the magnitude of this variance is defined by the corresponding eigenvalue. Consequently, PCA allows to reduce the dimensions number, with less loss of information. To compute the principal components, PCA operates according three steps which are the compute of the mean of each point cloud, the compute of the covariance matrix and finally the principal component determination using the eigenvectors of covariance matrix

- **Compute the mean of Point cloud U :**

In this step, we compute the mean of each dimension as follows :

$$\bar{u} = \frac{1}{n} \sum_{i=1}^n u_i. \quad (2)$$

- **Compute the covariance matrix:**

The covariance matrix denoted $CovU$ is computed from the outer product of matrix U with itself

$$CovU = \frac{1}{n} \sum_{i=1}^n (u_i - \bar{u})(u_i - \bar{u})^T \quad (3)$$

- **Compute the eigenvectors and eigenvalues of the covariance matrix:**

In this step, we compute the eigenvectors and eigenvalues for the covariance matrix $CovU$. First, we compute A : the diagonal matrix of the eigenvalues of $CovU$. Then, we search the matrix E of eigenvectors which diagonalizes the matrix $CovU$.

$$E^{-1}CovUE = A \quad (4)$$

The plane that we search was defined by the two first principal Components.

- **Project clouds U and V onto planes:** The principle of point projection is to project clouds U and V onto the planes defined above. The projected clouds preserve

the real cloud topology and the distribution of the entire model. This projection process is similar to observe a 3D point cloud from a certain viewing direction.

- **Apply NDT onto projected clouds:**

This step consists in applying NDT onto projected clouds.

V. EXPERIMENTAL STUDY

Our approach is evaluated using fifty 3D point clouds of different pieces and with different sizes. Since our approach is an improvement of the NDT algorithm, we have compared it to the original version of NDT. For a purpose of comparison, the Point Cloud Library (PCL) NDT implementation is used [14]. The NDT algorithm is implemented with a grid size of 1. The optimization is achieved when the gradient norm or the step size norm falls below 10E-6.

- **Data Sets**

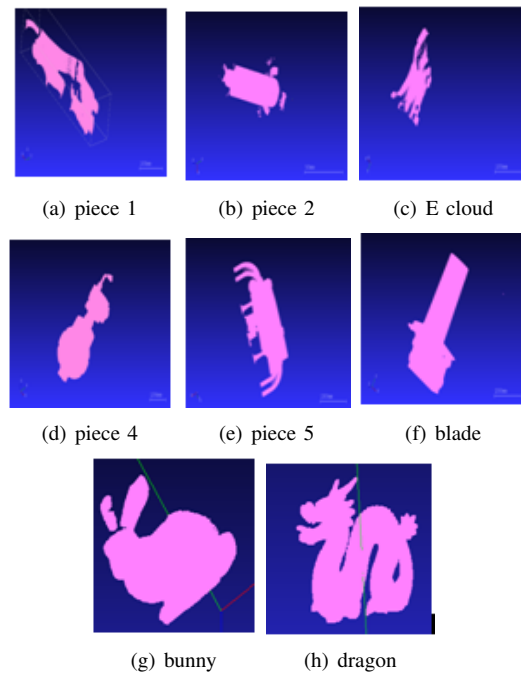


Fig. 1: 3D point clouds

In order to verify the feasibility and effectiveness of the proposed algorithm, experiments are carried out upon 50 3D point clouds. We use about 20 clouds with real transformations. The other clouds are obtained by transforming the original clouds through rigid transformations. In this paper, we prove the effectiveness of our method using the results of all tests. However, for clearness reasons, we present the other results for only eight clouds chosen to be variant and significant. These clouds (c.f Fig1) are whether downloaded from 3D Stanford scan repository [15] or generated from a CAD using software provided by GEOMNIA Company [16]. The clouds sizes vary between 45000 and 1765119 points.

A. Accuracy

In order to assess our approach, we measured the accuracy of NDT with original point clouds and NDT with projected point clouds. The accuracy is measured by the standard deviation (Stdv). This latter is computed as the sum average of the Euclidian distances of every corresponding point pair between the model and the reference point clouds.

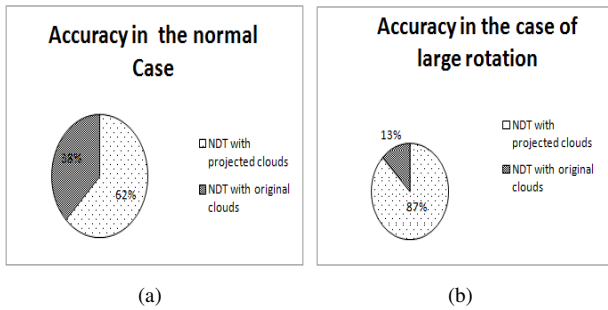


Fig. 2: (a) and(b): Accuracy comparison between NDT with projected clouds and original NDT

In order to have an overall view on the whole tests, we give in Fig.2 the number of times (in%) where NDT applied on projected clouds gives a best precision among all tested point clouds. The obtained results are 62% and 88% in the normal case and in the case of large rotation respectively. We conclude that our approach is more effective in the case of large rotation. For more details, we analyze in follows the experimental results obtained with the eight clouds of Fig.1.

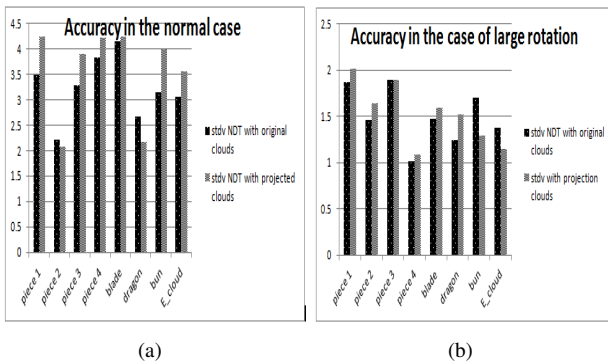


Fig. 3: (a) and(b): Accuracy comparison for eight clouds

For more clearness, we use a semi cologarithmic scale (i.e. the abscissa is unchanged and the ordinate is replaced by its cologarithm e.g. 10E-3 is represented by 3 to represent the standard deviation in the histogram). Note that the higher the value in the accuracy histogram is the more accurate the algorithm is.

Fig.3 shows the accuracy comparison of NDT applied on original clouds and NDT applied on projected clouds in the normal case and in the case of large rotation. In most cases,

NDT with projected clouds gives the best precision. This returns to the fact that the Gaussian distribution of each voxel well models surfaces. This proves that the proposed method is effective for the registration of point clouds in the two cases (normal and large rotation).

B. Execution Time

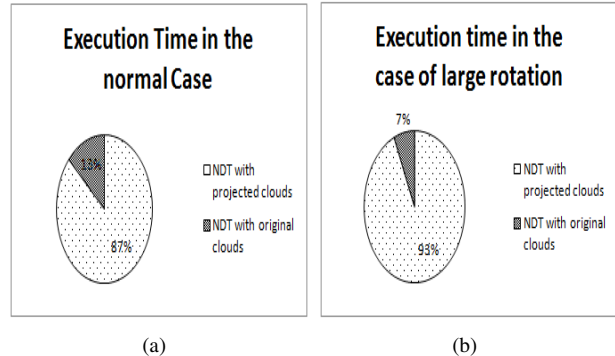


Fig. 4: (a) and(b): Execution time comparison between NDT with projected clouds and original NDT

Fig.4 shows the number of times (in%) where NDT applied on projected clouds runs faster than NDT applied on original clouds. We see that in the most cases our proposition runs faster than NDT when applied on original clouds. This is due to the fact that computing of normals for planar surface is less expensive in terms of time.

To show the result obtained with the clouds of Fig.1, we

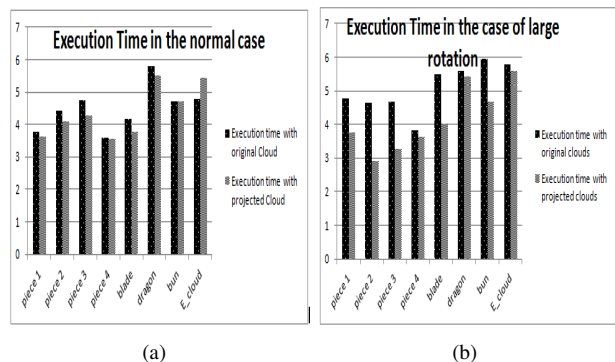


Fig. 5: (a) and(b): Execution Time comparison for eight clouds

use a semi logarithmic to represent the execution time for the eight clouds (cf Fig.5). Note that the lower the value is in the execution time histogram, the better the algorithm is. Referenced to Fig.5, the execution time of NDT applied on projected clouds is more reduced compared to NDT applied on original clouds.

VI. CONCLUSION

This work presents an approach to improve the execution time and accuracy of NDT. Since NDT gives better results when applied to planar surfaces, we propose to transform the two original clouds onto planar clouds. This is done using the PCA method to define a suitable plane in which each original cloud will be projected. The experimental study, we have carried out, shows that our approach improves the convergence basin, as well as it improves the computation time. Many directions are the subject of future work. We want intend mainly to remove discontinuities in the cost function caused by the cell boundaries found in NDT with projected cloud.

ACKNOWLEDGEMENTS

We are grateful to our professor Zaher MAHJOUB for his academic support and Dr Amine Kamoun, manager in the company Geomnia for his help.

REFERENCES

- [1] M. Attia, Y. Slama, and M. A. Kamoun, "On performance evaluation of registration algorithms for 3d point clouds," in *International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, 2016, pp. 45–50.
- [2] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.
- [3] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, 2009.
- [4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision*, vol. 2. IEEE Computer Society, 1999.
- [5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, pp. 346 – 359, 2008.
- [6] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3. IEEE, 2003, pp. 2743–2748.
- [7] R. Fletcher, *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, 1987.
- [8] A. Das, J. Servos, and S. L. Waslander, "3d scan registration using the normal distributions transform with ground segmentation and point cloud clustering," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 2207–2212.
- [9] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations," *The International Journal of Robotics Research*, vol. 31, pp. 1377–1393, 2012.
- [10] C. Ulaş and H. Temeltaş, "3d multi-layered normal distribution transform for fast and long range scan matching," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 1, pp. 85–108, 2013.
- [11] J. W. Kim and B. Lee, "Robust and fast 3-d scan registration using normal distributions transform with supervoxel segmentation," vol. -1, pp. 1–29, 2014.
- [12] M. Attia and Y. Slama, "Efficient initial guess determination based on 3d point cloud projection for icp algorithms," in *International Conference on High Performance Computing and Simulation HPCS*, 2017.
- [13] W. S. Yambor, B. A. Draper, and J. R. Beveridge, "Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures," in *Empirical Evaluation Methods in Computer Vision*. World Scientific Press, 2002.
- [14] "Pcllib," <http://pointclouds.org/downloads/>.
- [15] "3d repository scanner," <https://graphics.stanford.edu/data/3Dscanrep/>.
- [16] [Online]. Available: www.geomnia.eu