

Algorithm for the Integration/Registration of Three-Dimensional Point Clouds Based on Deep Learning Techniques

Juan R. Albarracín, Ángel F. Gómez, Andrés L. González, Jaime E. Meneses

juan.albarracin1@correo.uis.edu.co, angel.gomez1@correo.uis.edu.co, andres2178179@correo.uis.edu.co, jaimen@uis.edu.co

Universidad Industrial de Santander, Bucaramanga, Colombia

ABSTRACT:

With the continuous development of three-dimensional (3D) reconstruction techniques and the increase in the applications in which this technology can have a place, many methodologies have emerged that try to solve the problem of registering and aligning point clouds located in random positions or when they do not have an estimate of the transformation that differentiates them [3]. In this three-dimensional reconstruction process, after scanning or obtaining data from different points of view, one of the main challenges is establishing the corresponding points between the different captures to correctly align that pair of point clouds (integration or registration) based on the same reference system. Is necessary adjust and align point clouds corresponding to scans of the object or scene in question to perform the reconstruction or digitization for whatever application. In this research work, an algorithm that involves Deep Learning techniques is proposed to obtain a correct registration of point clouds captured in occlusion environments, high reflectance, and missing data.

POINT CLOUD REGISTRATION:

The main problem is to find the rigid transformation T (specifically find the rotation parameters R and translation t for a fixed point-set P_f , to a displaced point-set P_d , according with 1,

$$P_f = T \{ P_d \cdot R + t \}, \quad (1)$$

POINT CLOUD COMPARISON: DPDIST

To establish the correspondence of points, in the first instance, a way to compare the point clouds is needed, for this we use Deep Point Cloud Distance (DPDist) [1], which represents the point clouds in the form of a modified fisher vector 3DmFV [2], and calculates a quantitative metric between the two point clouds using a convolutional neural network.

[X, Y, Z] to 3DmFV representation:

With DPDist the representation of the coordinate system is changed to a two-dimensional matrix, which through statistical values describes the distribution of points in space according to a Gaussian grid (Figure. 1).

The 3DmFV uses 20 statistics calculated for each gaussian found from a Gaussian Mixed Model (GMM), to represent the points in terms of their distribution within the Gaussian grid, according with 2. (Figure. 2).

$$G_{3DmFV}^x = \begin{bmatrix} \sum_{t=1}^T L_{\lambda} \nabla_{\lambda} \log u_{\lambda}(p_t) |_{\lambda=\alpha, \mu, \sigma} \\ \max_t (L_{\lambda} \nabla_{\lambda} \log u_{\lambda}(p_t)) |_{\lambda=\alpha, \mu, \sigma} \\ \min_t (L_{\lambda} \nabla_{\lambda} \log u_{\lambda}(p_t)) |_{\lambda=\mu, \sigma} \end{bmatrix} \quad (2)$$

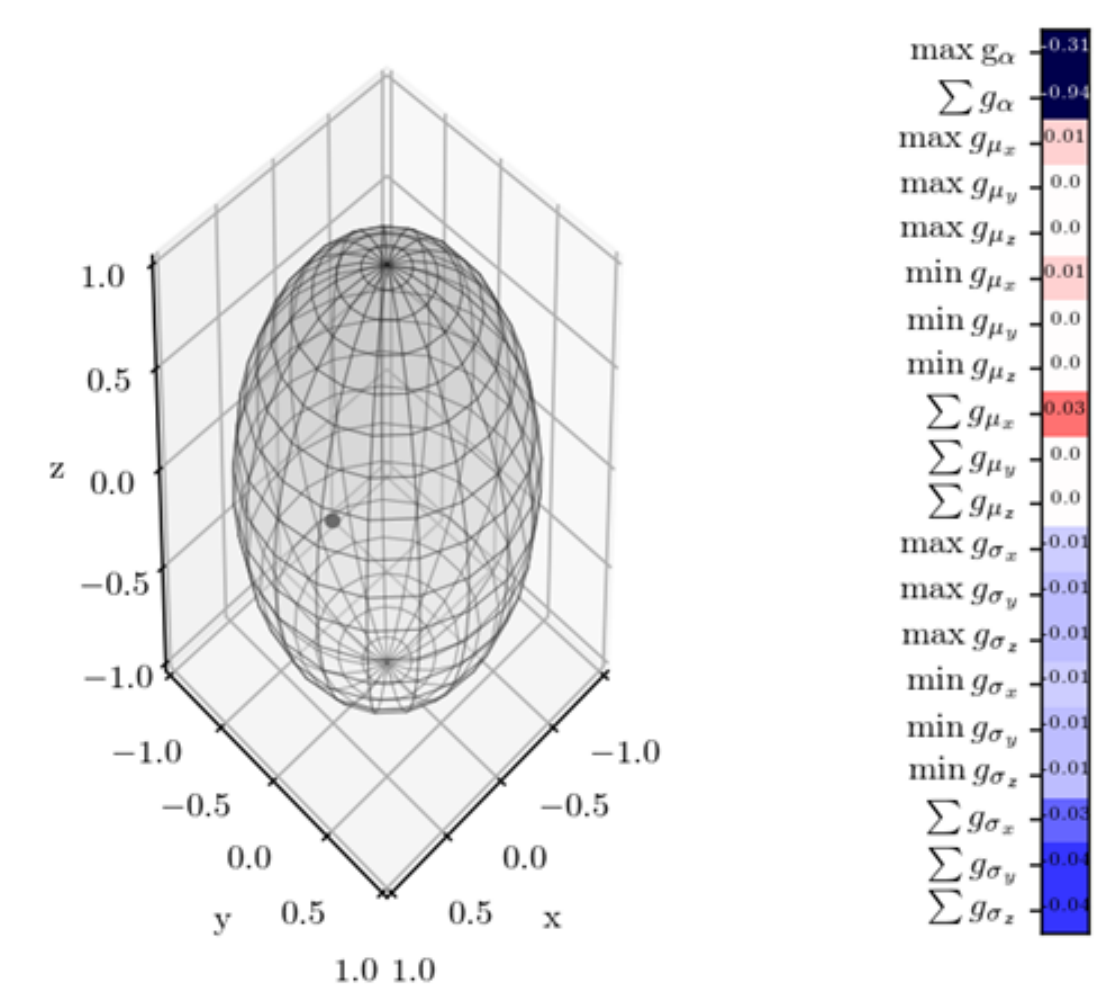


Figure 1. 3DmFV representation for only one gaussian. [2]

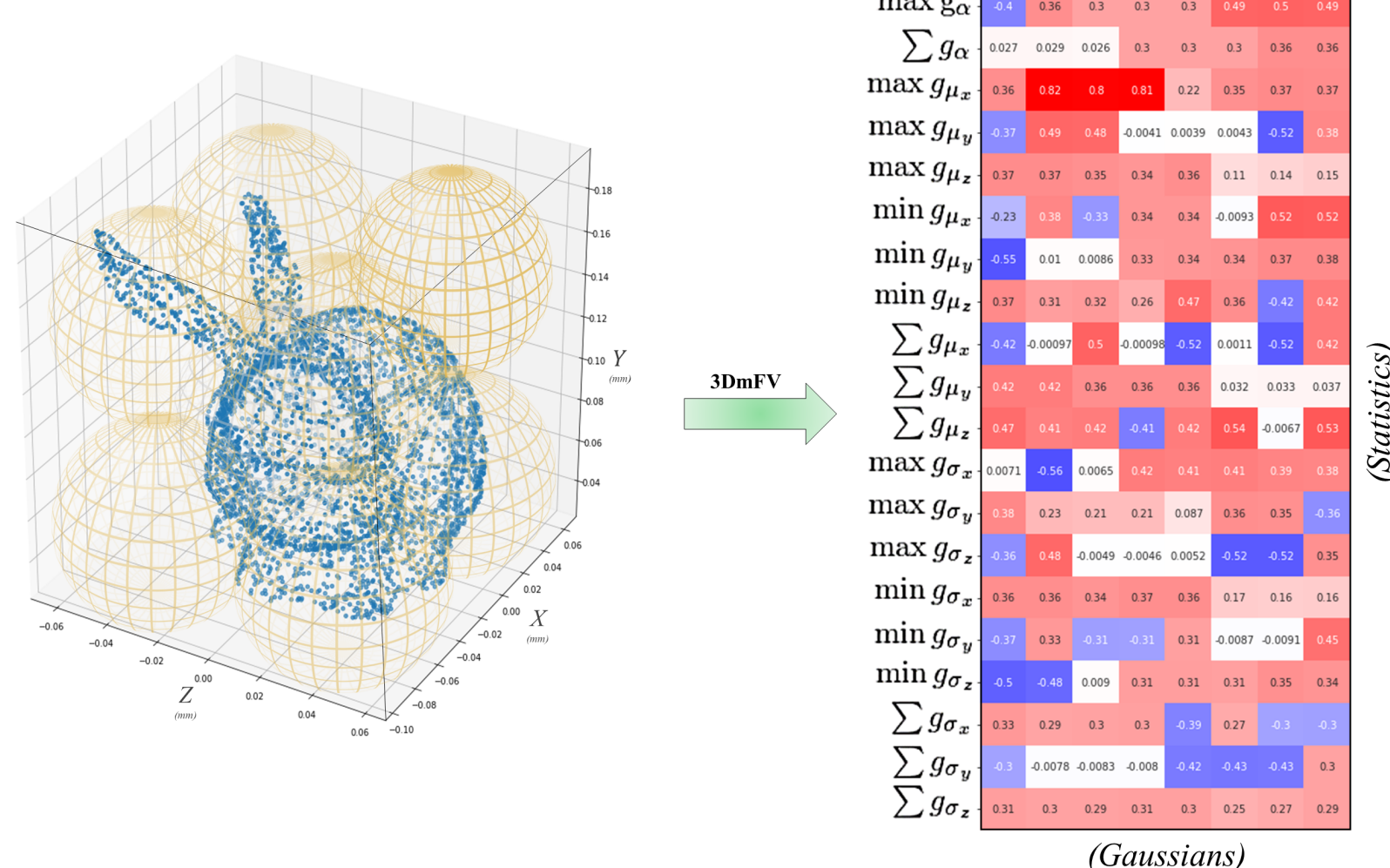


Figure 2. Representation of a point cloud using the 3DmFV (8 gaussians).

Underlying Surface:

To compare two sets of points A and B taken from the same point cloud, but without any exact point coincidence, 3DmFV generates an underlying surface from the set A, allowing the comparison distance DPDist between the points set B and the generated underlying surface

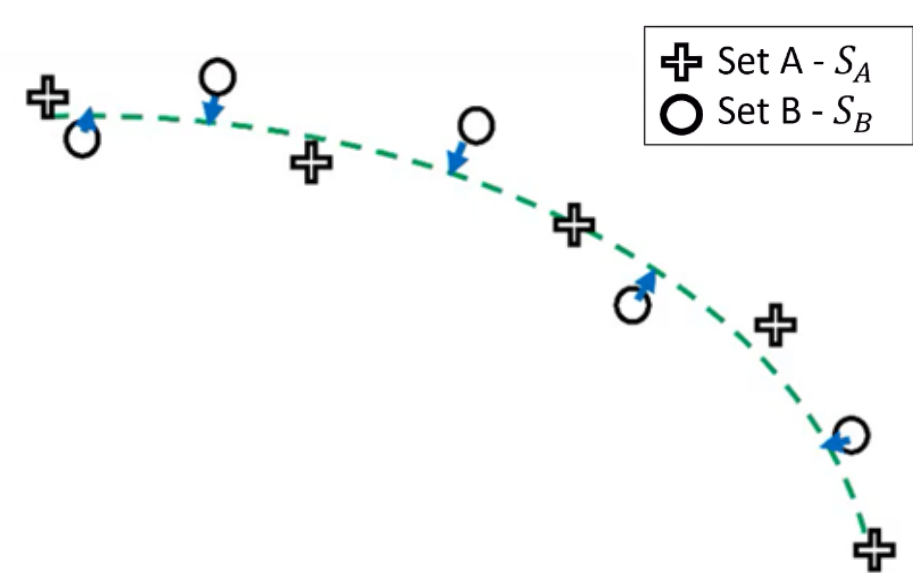


Figure 3. Underlying surface generated from a set of points. [1]

DPDist's Architecture:

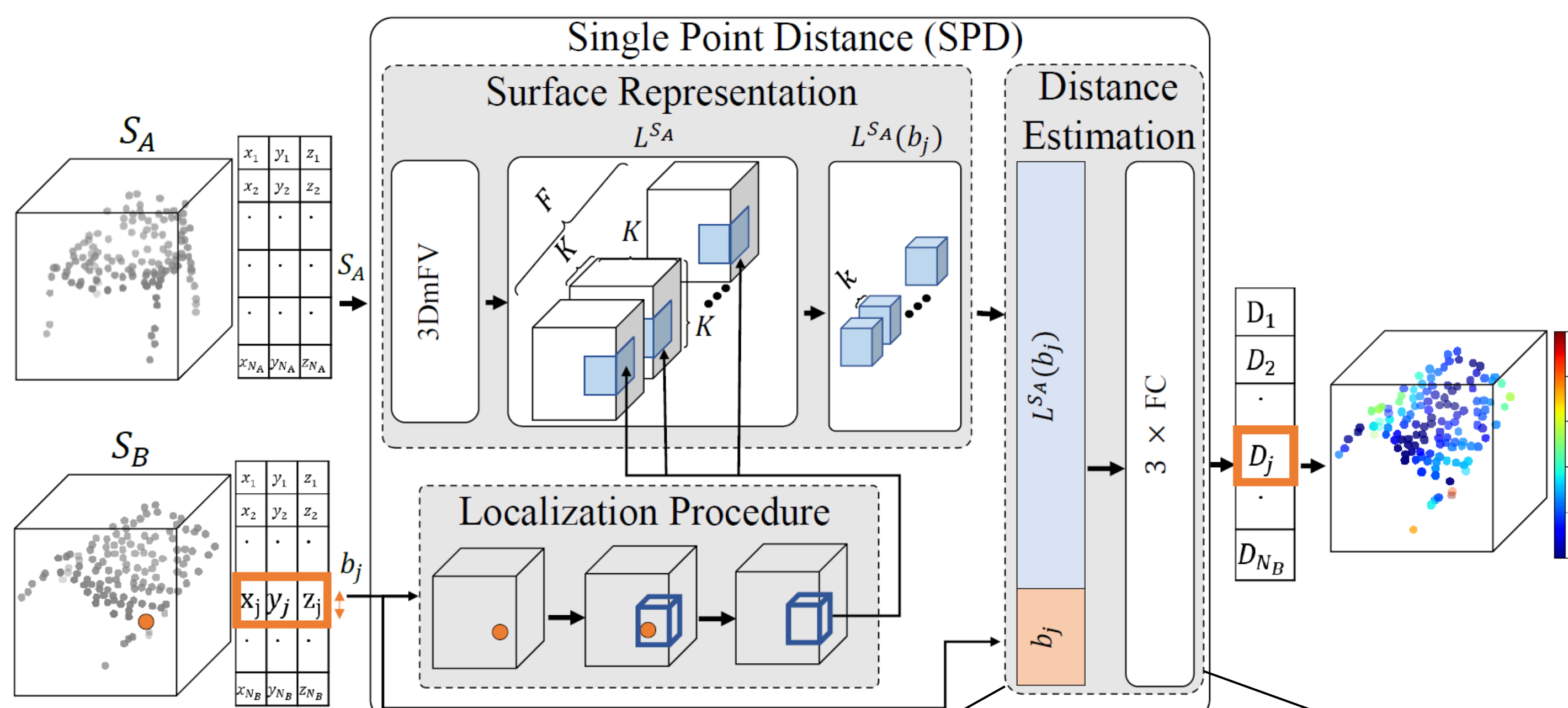


Figure 4. DPDist structure to estimate the distance from a point to surface (SPD). [1]

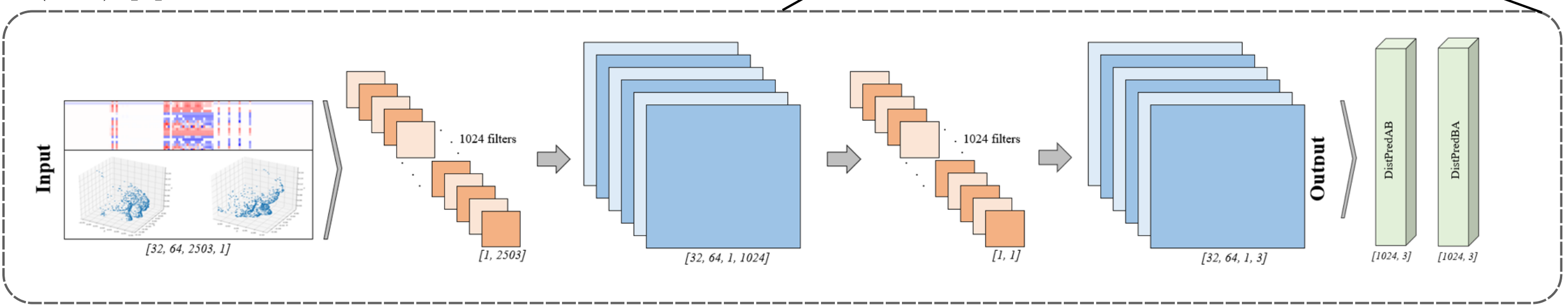


Figure 5. Convolutional Neural Network Architecture that estimates DPDist distances.

DPDIST GRADIENTS TO FIND CORRESPONDENCES

As DPDist intrinsically locates the points whose comparison distance value is minimum, it was proposed to obtain the direction vector from the gradient at each point using the backpropagation of the model to find the correspondences of points when performing the automatic differentiation between the prediction and the input data.

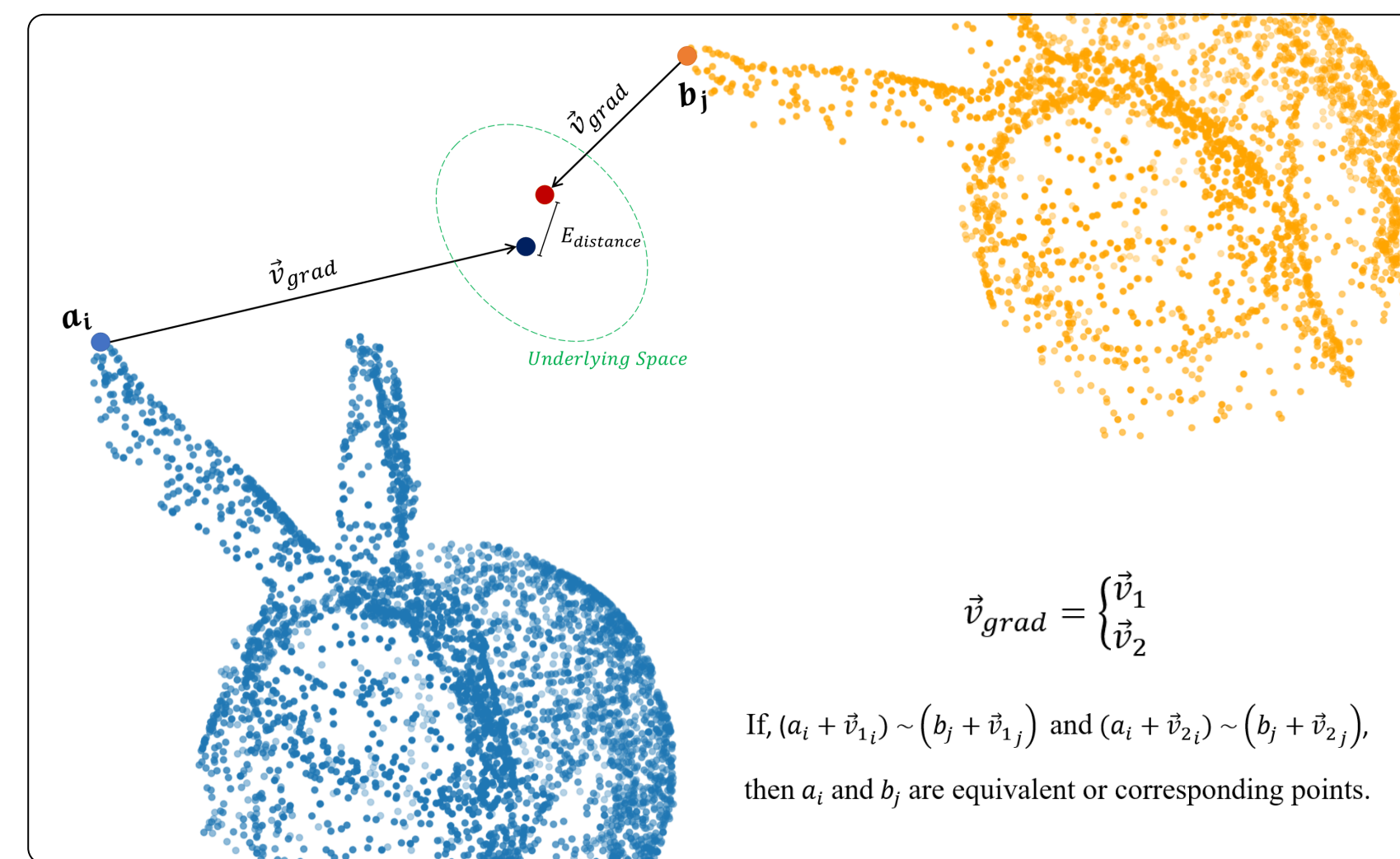


Figure 6. Points correspondence from the sum of the gradient vectors.

As DPDist generates an underlying surface of each cloud, the sum of the gradient found for each point of point cloud A and point cloud B does not lead directly to the coincident in the other cloud, instead it leads to an underlying space where the pairs of corresponding points of A and B will be found. The closest pairs of points in this new space are considered coincident. To determine their proximity, the Euclidean Distance was used.

ALGORITHM

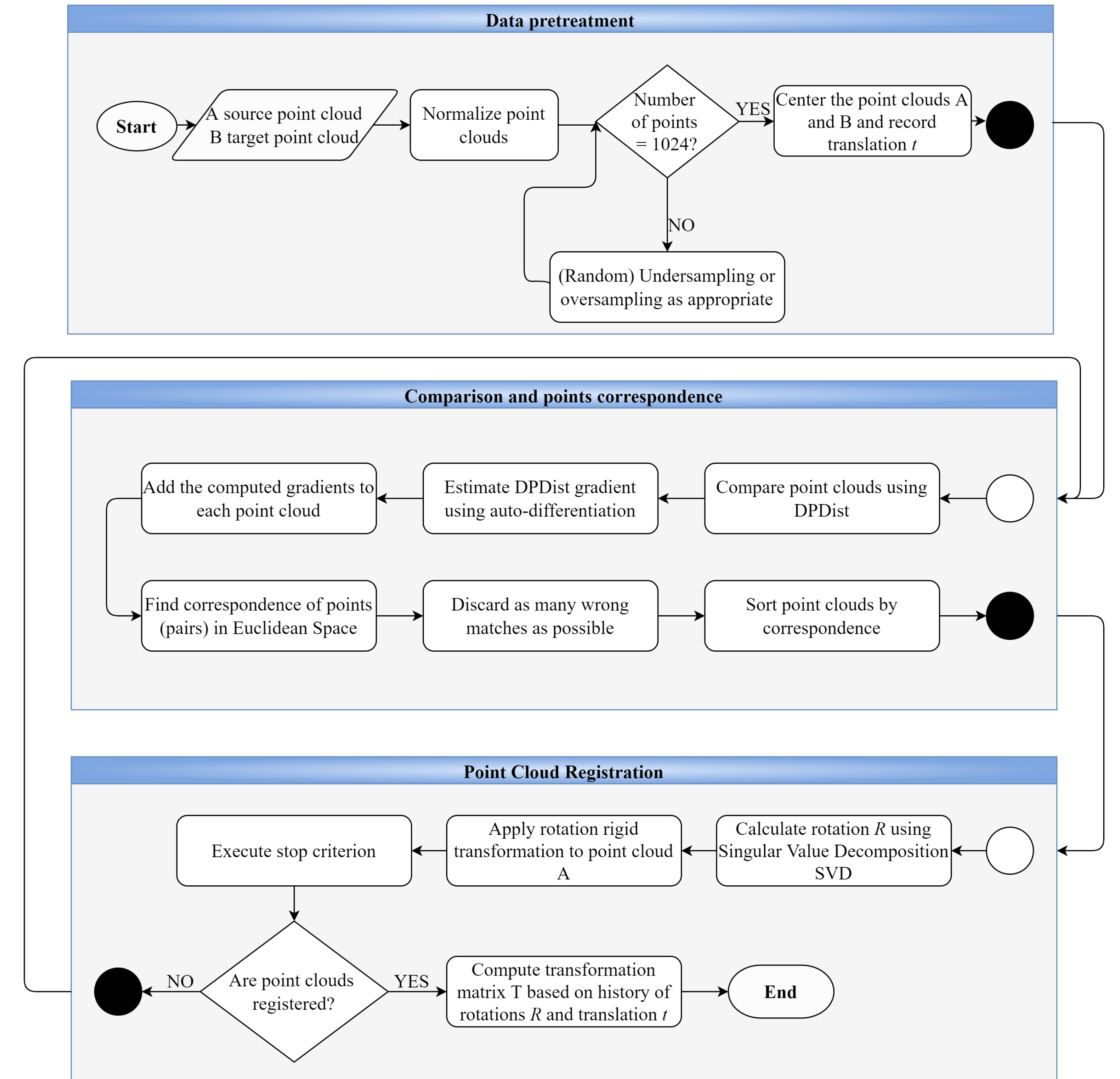


Figure 7. Flowchart of the proposed Algorithm.

RESULTS

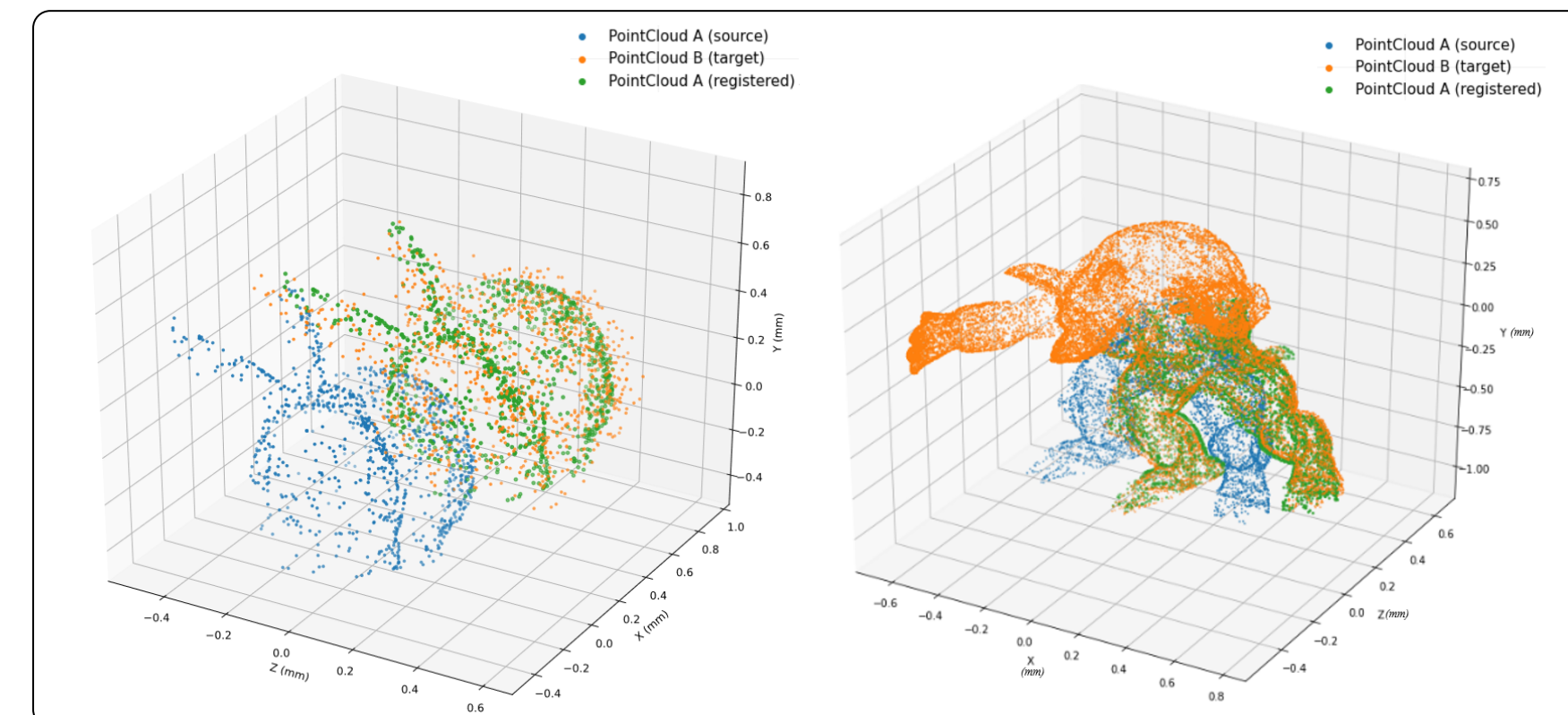


Figure 8. Point cloud registration examples using the proposed algorithm.

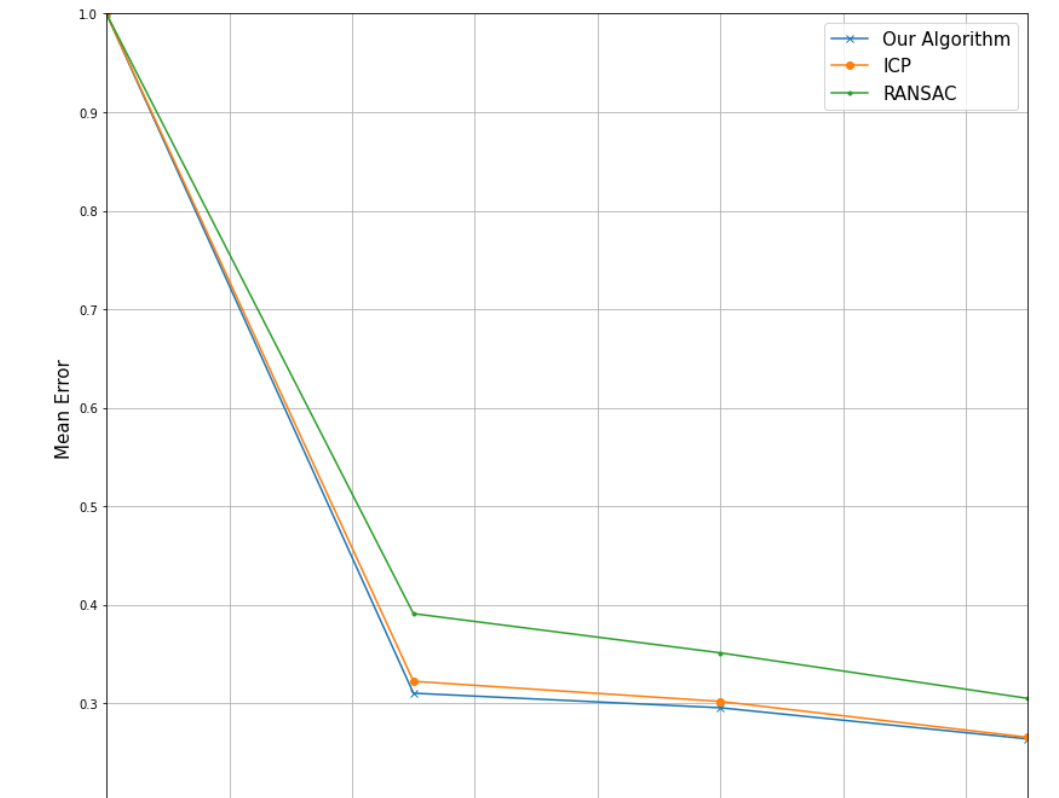


Figure 9. Mean Error comparison by iterations obtained for different algorithms.

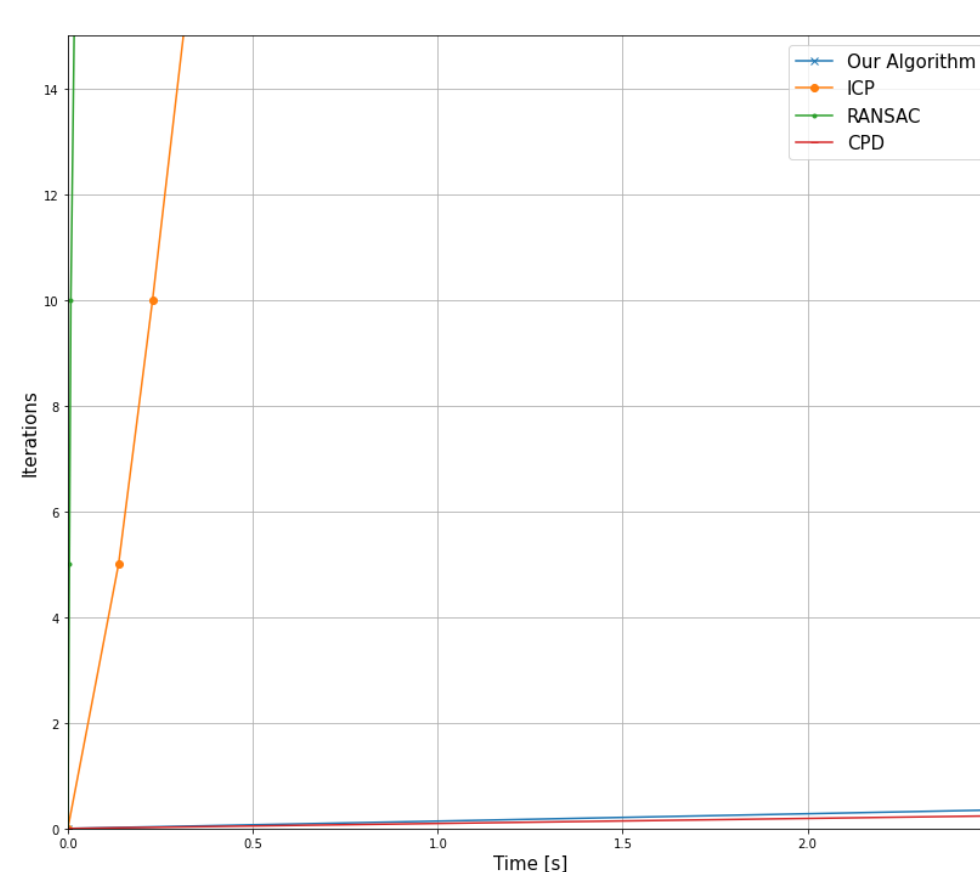


Figure 10. Execution time comparison by iterations obtained for different algorithms.

The proposed algorithm does not need an initial estimation of correspondence of points, but it automatically estimates the pairs of coincident points in the two point clouds to be registered. The proposed algorithm sacrifices execution time, since based on the results, it is very slow compared to other algorithms such as ICP, RANSAC or CPD. The long times obtained in the proposed algorithm are due to the computational cost for the calculation or estimation of the gradients through automatic differentiation in each iteration of the algorithm.

REFERENCES

- [1] Urbach, Dahlia, Yizhak Ben-Shabat, and Michael Lindenbaum. "DPDist: Comparing point clouds using deep point cloud distance." European Conference on Computer Vision. Springer, Cham, 2020.
- [2] Ben-Shabat, Y., Lindenbaum, M., Fischer, A.: 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. IEEE Robotics and Automation Letters 3(4), 3145-3152 (2018)
- [3] B. Maiseli, Y. Gu, and H. Gao, "Recent developments and trends in point set registration methods," Journal of Visual Communication and Image Representation, vol. 46, pp. 95 – 106, 2017.