Neural network in the truncated point clouds registration problem

ID 350

A. Makovetskii[1], V. Kober[1], D. Zhernov[1] and A. Voronin[1]

[1] Mathematics Department of Chelyabinsk State University, Russian Federation

ITNT-2022

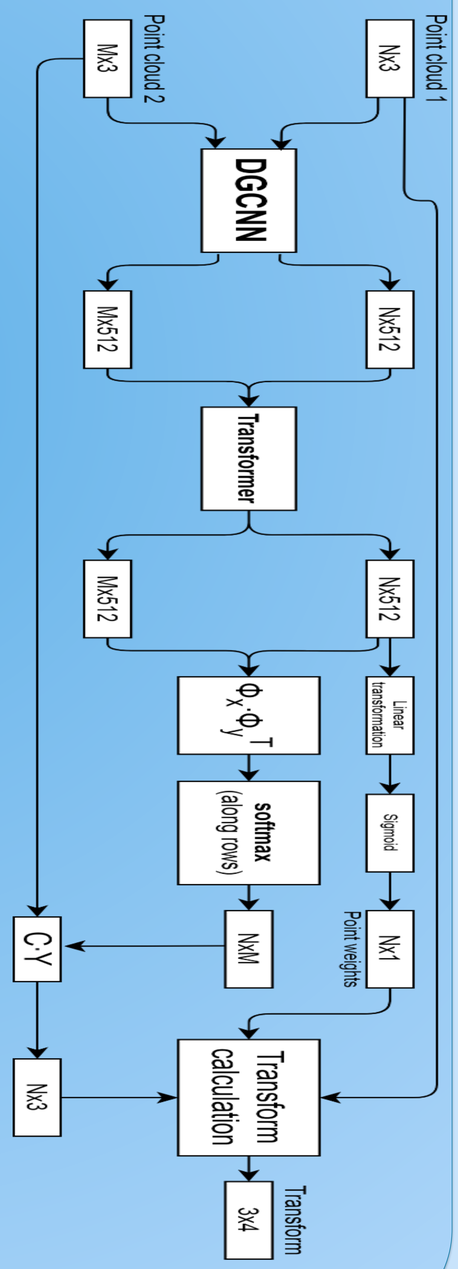
Point cloud registration is aimed at finding an orthogonal transformation to align one-point cloud with another. Such registration is a fundamental task in computer vision and robotics and is widely used in different applications, including 3D reconstruction, simultaneous localization and mapping, and autonomous driving. The most known approach for point clouds registration is Iterative Closest Points (ICP) algorithm. Recently, registration methods have begun to use deep learning. To exploit different aspects of the problem, various types of deep learning-based point cloud registration algorithms have been proposed. Particularly, the Deep Closest Points (DCP) network was described. In the proposed paper we apply a modified version of DCP to the incongruent point clouds registration task. We show by computer simulation the performance of the proposed network on ModelNet40 database.

The Neural Network Architecture

The network input data are two point clouds, $\boldsymbol{P} = \{\boldsymbol{p_1}, \dots, \boldsymbol{p_m}\}$, $\boldsymbol{Q} = \{\boldsymbol{q_1}, \dots, \boldsymbol{q_n}\}$, $\boldsymbol{p_i}, \boldsymbol{q_j} \in \mathbb{R}^3$, $\boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{m}$, $\boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{n}$. Output data is an orthogonal transformation in homogeneous coordinates.

The resulting network uses the DGCNN. for the initial extraction of the features of the point in the clouds, The DGCNN network, which is part of the DCP neural network, has four consecutive EdgeConv blocks that calculate a set of features for each point based on its neighborhood. The number of features calculated in EdgeConv blocks is 64, 64, 128, and 256, respectively. Further, the features obtained after each of the four blocks are merged into one set of features, to which a common pointwise linear transformation is applied. As a result, 512 features are obtained for each point. For each point. are applied the batch normalization and the LeakyReLU activation function. DGCNN is applied to each cloud independently.

Joint consideration of both point clouds gives the network the ability to compare points between different clouds, trying for each point of the first cloud to predict whether this point is in the second cloud or not. The architecture of the Transformer network used in this modification of the DCP is identical in all respects to the architecture of the Transformer from the original DCP. As a result of the described operations, a descriptor consisting of 512 numbers is calculated for each point of each cloud. The first cloud descriptors are used to compute the weight of a point (i.e. evaluate whether there is a point in the second cloud corresponding to a given point in the first cloud). This is a change from the original DCP architecture. The weight data is calculated by applying to each point of the first cloud an independent linear transformation with a sigmoid activation function. Also, as in the original DCP, the point descriptors of both clouds are used to find a correspondence between points in P and Q. The cross-covariance matrices for each point are multiplied by the neural network's predicted weight for this point. Here there is the difference between the proposed network and the original DCP.



The number of trainable weights of the resulting network is 5787265.

The error function is MSE between the estimated and true transformation matrices.

The Adam algorithm with the learning rate parameter is 0.001. The batch size is 4. The noise was not used during training. The network was trained in 50 epochs. During the training process, we gradually increase the maximum angle of rotation from 30 degrees to 180 degrees.

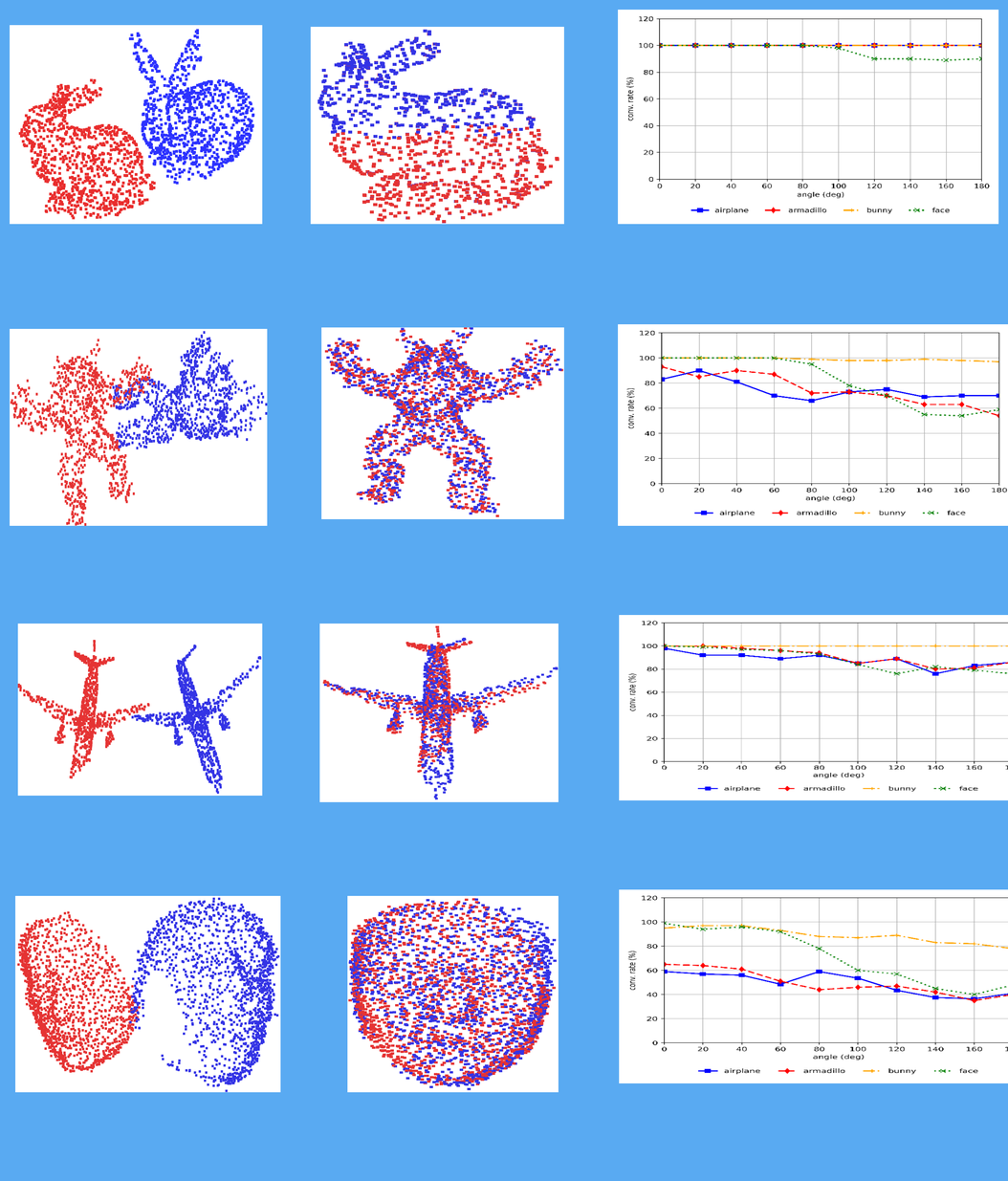
Computer simulation

Here we use the following four-point clouds: Stanford Bunny and Armadillo; airplane from the ModelNet40 database [19]; human face from the Bosphorus database [20]. The clouds of the Stanford Bunny, Armadillo, and airplane are composed of 1024 points, the face clouds are composed of 1724 points. All points lie in the sphere of the central unit. We apply the proposed neural network to the ModelNet40 database. The database contains 12311 mesh CAD models belonging to 40 classes. 80 percent of the models are used for training and 20 percent for testing. For each model, 1024 points are evenly sampled from the mesh faces. Only database information about the point coordinates is used.

We apply a rigid geometrical transformation to the point cloud P defined by the orthogonal matrix R_{true} and translation vector T_{true}. The point cloud Q is obtained from the point cloud P as

$$Q = R_{true}P + T_{true}.$$

Information about the matrix R_{true} and the translation vector T_{true} is contained in the matrix M_{true} of inhomogeneous coordinates.



References

- F Pomerleau, F .Colas, and R. Siegwart, “A review of point cloud registration algorithms for mobile robotics,” Foundations and Trends in Robotics, vol. 4(1), pp. 1–104, 2015.
- N. Fioraiox, and K Konolige. “Realtime visual and point cloud slam. In: Proceedings of the RGB-D workshop on advanced reasoning with depth cameras at robotics:” 2011.
- W. Lu, G. Wan , Y .Zhou, F. Xu, P. Yuan, and S. Song, “Deep VCP: an end-to-end deep neural network for point cloud registration,” ICCV, IEEE, pp. 12–21, 2019.
- Y. Wang, J. Solomon, “Deep closest point: learning representations for point cloud registration,” International Conference on Computer Vision (ICCV), IEEE, pp. 3522–3531, 2019.
- P. Besl, and N. McKay, “A method for registration of 3-D shapes,” IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 2, pp. 239-256, 1992.