STAG: Smart Tools and Applications in Graphics (2021)
P. Frosini, D. Giorgi, S. Melzi, and E. Rodolà (Editors)

Evaluating Deep Learning Methods for Low Resolution Point Cloud Registration in Outdoor Scenarios

[1] Arslan Siddique, [2] Massimiliano Corsini, [3] Fabio Ganovelli, [4] Paolo Cignoni
[1] arslan.siddique@isti.cnr.it, [2] massimiliano.corsini@isti.cnr.it, [3] fabio.ganovelli@isti.cnr.it, [4] paolo.cignoni@isti.cnr.it
[1,2,3,4] Visual Computing Lab, Instituto di Scienza e Tecnologia dell'Informazione, CNR, Pisa, Italy

Abstract

Point cloud registration is a fundamental task in 3D reconstruction and environment perception. We explore the performance of modern Deep Learning-based registration techniques, in particular Deep Global Registration (DGR) and Learning Multi-view Registration (LMVR), on an outdoor real world data consisting of thousands of range maps of a building acquired by a Velodyne LIDAR mounted on a drone. We used these pairwise registration methods in a sequential pipeline to obtain an initial rough registration. The output of this pipeline can be further globally refined. This simple registration pipeline allow us to assess if these modern methods are able to deal with this low quality data. Our experiments demonstrated that, despite some design choices adopted to take into account the peculiarities of the data, more work is required to improve the results of the registration.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Perception;***

1. Introduction

Point cloud data is becoming increasingly important to represent the 3D real world due to a significant decrease in the cost of 3D scanning sensors, especially those designed for automotive industry like Velodyne LIDAR and other devices designed for games and virtual presence, such as for example, Kinect (structured light IR) and Zed camera (stereo matching). However, the 3D scanning sensors capture the data only in a limited view range. Hence, the desired object needs to be scanned from different views and then an algorithm is required to align all the scans into a complete object or 3D scene. This process is called 3D registration. It is a fundamental task in 3D reconstruction and 3D perception for robots and autonomous cars. The aim of 3D registration is to estimate the transformation to align a reference point cloud with another one. By applying the transformation matrices progressively to all the scans of the same scene, a 3D object/scene is built. Typically, the registration error accumulates during these transformations and a global registration is required at the end.

A pairwise point cloud registration consists of four main steps. The first step is to extract feature points that are notable, in some aspect, in both scans. This step has been widely studied in literature and it can be done through hand-crafted features like spin images [JH99] as well as deep learning based features like Fully Convolutional Geometric Feature (FCGF) [FA21]. The second step is to match feature points and build correspondences between the features of both scans. The third step is to separate inliers and outliers in the big pool of correspondences. This is done mostly using RANdom SAmple Consensus (RANSAC) [AMCO08,SWK07]. The last step is to finalise the alignment considering the whole surface and estimating the final transformation matrix.

In this paper, we explore the performance of modern state of the art Deep Learning based point cloud registration methods on challenging data acquired in the ambit of the ENCORE project (`http://encorebim.eu/`). This project is devoted to create new tools and instruments to support the renovation process of existing building aimed to improve energy efficiency and comfort parameters. One of the part of the project involves rapid acquisition of existing buildings. The data have been acquired using a Velodyne LIDAR mounted on a drone, the range maps are collected in outdoor scenario. Such data consists of a large number of range maps acquired sequentially during the flight of the drone around a building. We applied Deep Global Registration (DGR) [CDK20] and Learning Multi-View Registration (LMVR) [GZW*20] for the processing and registration of these sequential range maps because these are both global registration methods which have achieved state of the art results on real world datasets e.g 3D Match Benchmark [ZSN*17], ScanNet [DCS*17] etc. and the conventional registration approaches fail to deliver any result on our dataset.

Figure 1 shows the visualisation of a part of the acquired range maps in MeshLab [CRC*11] and give a better understanding of our research problem. Even if the sequence of range maps is pre-aligned, the pre-alignment is not so good, for example we can observe mis-alignments in some regions. Our final goal is to reduce the registration error of these range maps. Here, since we are in the

DOI: 10.2312/stag.20211489

we adopt a simple registration scheme, described in Section 3. This approach requires a global registration step at the end, but is simple and it is useful to understand the improvement of the registration given by using modern Deep Learning-based registration methods.

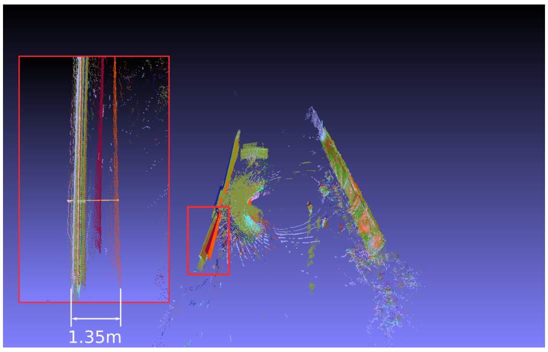

Figure 1: *A short sequence of range maps acquired by a drone equipped with a velodyne. Although this sequence is pre-aligned there are large margins of improvement. The zoom-in shows that the surface of the wall appears to be over one meter 'thick' in some regions, demonstrating the presence of significant registration errors.*

2. Related Work

Point cloud registration methods can be broadly classified either as a local registration or a global registration. Early works introduced local registration methods which work only if the transformation between the point clouds is small and the area of overlap between the point clouds is large. One of the most popular local registration algorithms is Iterative Closest Point (ICP). Three similar papers related to this algorithm were published in the same year of 1992. The most basic algorithm is point to point ICP [BM92] which builds correspondence for each point using Euclidean distance and computes transformation such that the distance between the correspondent points is minimized. Such an objective function makes it very sensitive towards outliers. Chen and Medioni [CM91] specifically considered the problem of aligning data for object modeling. Their approach considers model range map data to be planar while sensor range map data to be points and hence, introduces a point to plane variant of ICP. Zhang article considered a similar approach to that of [BM92] and also introduced a robust method of outlier rejection in the correspondence phase of the algorithm. A more generalized version is plane to plane ICP [SHT09] that considers both scans to be locally planar and is more robust to incorrect correspondences. Plane to plane ICP outperforms standard ICP and point to plane ICP while maintaining the speed and simplicity of other ICP algorithms.

Early work in the area of global registration comes from Li and Hartley [LH07] who combined Branch and Bound optimization [BMM99] with Lipschitz optimization [HJ95]. PointNetLK modifies the classical Lukas & Kanade (LK) algorithm [LK*81] on top of the PointNet [QSMG17] classification model and unrolls both of them into a single recurrent trainable neural network for the point cloud registration task. DeepICP [LWZ*19] is an end-to-end deep neural network for 3D point cloud registration. The network firstly generates correspondences using learned matched probabilities and then creates an aligned point cloud. 3DRegNet [PRG*20] presented a Procrustes [Gow75] approach using SVD to estimate the transformation. RPM-Net [Gow75] extracts hybrid features learned from both spatial coordinates and local geometry and further introduces a secondary network to predict optimal annealing parameters.

3. Methodology

The paper aims to present our experimental results of modern Deep Learning based point cloud registration methods on the challenging outdoor real-world dataset previously mentioned. In the following, we describe the two methods used in our tests and the registration pipeline which exploits them to assess their performance in this scenario.

3.1. Deep Global Registration

Deep Global Registration (DGR) is a differentiable framework for pairwise registration of 3D scans. First of all, it extracts deep learning based Fully Convolutional Geometric Features (FCGF) [FA21] for both scans. Points which do not satisfy transformation equation are filtered out. Then it uses a U-Net structure that consists of residual blocks between strided 6D convolutions to predict correspondence probabilities. The U-Net structure formulates inliers as a foreground segmentation problem and separates inliers correspondences from outliers. Then a weighted Procrustes method minimises the translation and rotation errors to give a transformation matrix. It assigns a weight of zero to non-overlapped region which leads to a significant reduction in the computational complexity. Finally a fine tuning model reduces robust loss function in a globally consistent manner.

3.2. Learning Multi-View Registration

Learning Multi-view registration (LMVR) formulates the conventional two stage technique as an end-end neural network. During the forward pass, it estimates pairwise transformation parameters as well as the transformation synchronization. First of all, it extracts FCGF features of all points clouds and feeds these features into a softNN layer to calculate the correspondences between the point clouds. These correspondences are passed through a series of registration blocks. The initial registration block outputs the per-correspondence weights and initial transformation parameters. These initial weights and parameters are passed into the several iterations in a registration refinement. After each iteration, the estimated transformation parameters are used to pre-align the correspondences concatenated with the weights from previous iteration. The outputs from these iteration build a graph

3.3. Registration pipeline

The block diagram of the proposed pipeline is shown in Figure 2. This pipeline takes into account the fact that the range maps have been collected sequentially. The range maps have been acquired at a very high sampling rate and their initial position and orientation are quite close to each other. Hence, most of the methods consider the two consecutive range maps as the same. Therefore, in our pipeline, the range maps are processed using an interval of k range maps. We start with the pair i and $i+k$ and then proceed towards the range map $i+2k$, and so on.

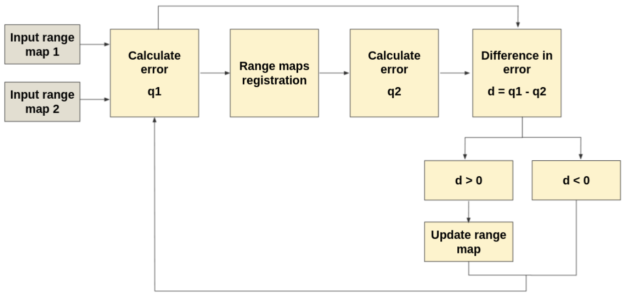

Figure 2: *Block diagram of the proposed methodology*

As a first step, we measure the alignment error between the two input range maps. Then pairwise registration is performed with one of the two networks in order to calculate the transformation matrix M. The second range map is transformed according to M to align with the first range map. At this point, we measure the alignment error again. If the post-registration alignment error is lower than the the pre-registration alignment error, it indicates a good registration and then we accumulate the two range maps into a single range map. However if, the post-registration alignment error is higher than the the pre-registration alignment error, the registration operation has made our alignment worse and hence, we do not update our range maps and discard the second range map. Finally, the pipeline is executed again with a new range map process. Hence, the range map 1 is always updated with the current results and the registration map 2 is the range map $i+hk$ where h increases during the processing of the sequence.

The next section details how the registration error is evaluated.

3.4. Alignment error metric

In order to evaluate the registration error, we use a simple error metric based on the distance between the two range maps weighted by the corresponding normals. Consider the scenario shown in Figure 3. Blue circles indicate one range map $\overrightarrow{P_a}$ and red circles indicate another range map $\overrightarrow{P_b}$.

Every range map consists of a cluster of 3D points. For each point, first, we find the nearest point in the other range map. The distance vector between the two points is calculated.

$$\mathrm{d} = \|\overrightarrow{P_a - P_b}\|^2 \, (\overrightarrow{N}_{P_a} \cdot \overrightarrow{N}_{P_b}) \tag{1}$$

where, $\overrightarrow{N}_{P_a}$ and $\overrightarrow{N}_{P_b}$ are normals to the points. The magnitude of this distance vector represents the alignment error. The normals are used to penalize the pairing of points belonging to different portions of the wall around the corners. This is particularly important for this type of dataset because corners, such as those on the contours of doors and windows and those formed by different walls are essentially the only features of the geometry. Note that this is especially true for LIDARs such as Velodyne, designed for the automotive and similar applications and hence with a relatively poor sampling density.

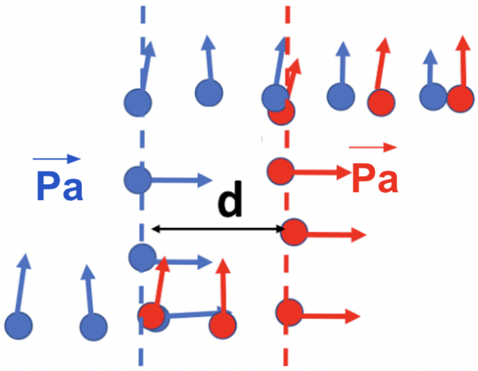

Figure 3: *Illustration of two range maps to measure alignment error.*

For this reason the error calculation is performed only considering the region around the corners, which are found by segmenting the point cloud in planar regions and then finding the geometric intersection of the adjacent ones (see [CSAD04]).

We expect this distance error between the range maps to decrease after point cloud registration.

4. Implementation and Experimental Setup

As pointed in the Section. 1, we are applying point cloud registration techniques on the ENCORE dataset. The data consists of 7290 point clouds sampled in a sequential manner using a Velodyne LIDAR. DGR and LMVR, both require Minkowski Engine [CGS19] to be installed. Open3D [ZPK18] library is used for reading, visualising and processing point cloud files. Installing correct versions of the libraries and CUDA is very important to avoid errors. Our experimental setup consists of Ubuntu 18 running Python 3.8, CUDA 10.2, cuDNN 7.6.5, Open3D version 0.10.0, PyTorch 1.7.1 and Minkowski Engine 0.4.3. Installation of CUDA is mandatory for Minkowski Engine installation. Windows is not currently supported for Minkowski Engine. Hence, ubuntu and NVIDIA GPU are recommended for working with Minkowski Engine.

We used 8GB NVIDIA RTX 2070 graphics card and each point cloud has more than 10000 points. As previously explained, during the processing of the sequence of range maps, after a good alignment the two point clouds are accumulated into a single point cloud. With this approach we face memory error issue after few updates, but this is a minor problem since we want to assess the capability of the modern deep learning registration methods, and not to process the entire sequence.

5. Results and Discussions

We started our experiments with conventional registration methods and moved on to more advanced and complex approaches. The next sub-sections give details of their performances.

5.1. Conventional registration methods

We started our experiments by applying Open3D [ZPK18] implementations of ICP variants [BM92, CM91, SHT09] on our pre-aligned sequences. However, these approaches did not give us any good registration results. The transformation matrix turns out to be a unity matrix indicating that these methods consider the point clouds being already aligned. So we decided to evaluate more advanced and recent registrations methods on our data.

5.2. Deep learning based registration methods

We applied DGR and LMVR on our dataset and plotted the distance alignment error before and after registration. Both registration algorithms give very similar plots. Figure 4 shows alignment error before and after registration by applying LMVR on the data using a step of k = 50. The plot shows that at the start point clouds are well aligned and hence, distance error before and after registration is similar. But by applying LMVR, alignment of reaming point clouds is significantly improved and alignment error is reduced significantly after doing the registration.

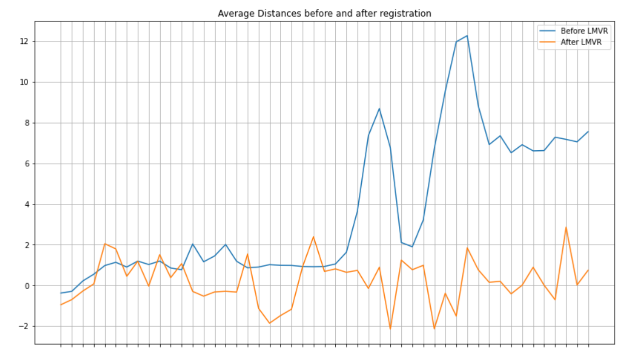

Figure 4: *Alignment error before and after doing the LMVR registration on a series of point clouds*

The registration algorithm gives good quality results if there is a large amount of intersection of point clouds. An example is shown in Figure 5 in which registration algorithm gives very well aligned point clouds.

At the same time, our experiments indicate that the two networks do not produce reliable results sometimes. Even by varying k, both DGR and LMVR fail sometimes to correctly align the input range maps. It can be due to lower intersection between the point cloud regions and the low resolution feature of the data. Our idea is to use our alignment error metric to be able to detect such poor quality registrations and discard them. Although the metric 1 is typically used in ICP to better evaluate the alignment in presence of corners, it works mostly when the range maps are reasonably aligned, otherwise this measure is not reliable. Hence when network fails the

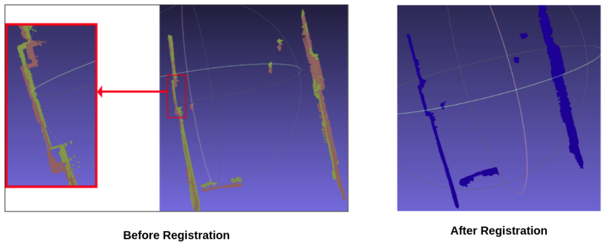

Figure 5: *An example of a good quality registration.*

registration, it gives completely random results, and hence the normal weights drive to a wrong evaluation of the registration quality. An example is shown in Figure 6; although the distance calculated using 1 between the range maps is reduced after DGR registration, it is easy to observe that the registration has failed.

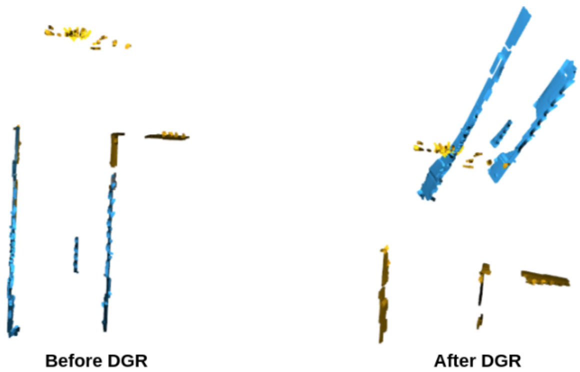

Figure 6: *An example of a poor quality registration in which our alignment error metric fails to consider it a poor quality registration*

6. Conclusions and Future recommendations

In this paper, we reported the preliminary results of the ongoing effort to explore the performance of modern deep learning based point cloud registration methods on an outdoor real world dataset. The quality of registration is poor because the scenario of our dataset is very different from the typical datasets on which these methods have been trained. Our dataset is challenging because of its low resolution, non-uniform distribution of the points, and other problems such as the presence of flat surfaces and corners. However, we believe that a better evaluation of the registration error will help us to correctly discard wrongly aligned pairs and drive to a dataset with a better rough registration.

References

[AMCO08] AIGER D., MITRA N. J., COHEN-OR D.: 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 1–10. URL: https://doi.org/10.1145/1360612.1360684, doi:10.1145/1360612.1360684. 1

[BM92] BESL P., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 239–256. doi:10.1109/34.121791. 2, 4

[BMM99] BEMPORAD A., MIGNONE D., MORARI M.: An efficient branch and bound algorithm for state estimation and control of hybrid systems. In *1999 European Control Conference (ECC)* (1999), pp. 557–562. doi:10.23919/ECC.1999.7099363. 2

[CDK20] CHOY C., DONG W., KOLTUN V.: Deep global registration. *CoRR abs/2004.11540* (2020). URL: https://arxiv.org/abs/2004.11540, arXiv:2004.11540. 1

[CGS19] CHOY C. B., GWAK J., SAVARESE S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. *CoRR abs/1904.08755* (2019). URL: http://arxiv.org/abs/1904.08755, arXiv:1904.08755. 3

[CM91] CHEN Y., MEDIONI G.: Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation* (1991), pp. 2724–2729 vol.3. doi:10.1109/ROBOT.1991.132043. 2, 4

[CRC*11] CIGNONI P., RANZUGLIA G., CALLIERI M., CORSINI M., GANOVELLI F., PIETRONI N., TARINI M.: Meshlab. 1

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 905–914. URL: https://doi.org/10.1145/1015706.1015817, doi:10.1145/1015706.1015817. 3

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T. A., NIESSNER M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR abs/1702.04405* (2017). URL: http://arxiv.org/abs/1702.04405, arXiv:1702.04405. 1

[FA21] FENG Q., ATANASOV N.: Fully convolutional geometric features for category-level object alignment. *CoRR abs/2103.04494* (2021). URL: https://arxiv.org/abs/2103.04494, arXiv:2103.04494. 1, 2

[Gow75] GOWER J. C.: Generalized procrustes analysis. *Psychometrika 40*, 1 (1975), 33–51. 2

[GZW*20] GOJCIC Z., ZHOU C., WEGNER J. D., GUIBAS L. J., BIRDAL T.: Learning multiview 3d point cloud registration. *CoRR abs/2001.05119* (2020). URL: https://arxiv.org/abs/2001.05119, arXiv:2001.05119. 1

[HJ95] HANSEN P., JAUMARD B.: Lipschitz optimization. In *Handbook of global optimization*. springer, 1995, pp. 407–493. 2

[JH99] JOHNSON A., HEBERT M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 5 (1999), 433–449. doi:10.1109/34.765655. 1

[LH07] LI H., HARTLEY R.: The 3d-3d registration problem revisited. In *2007 IEEE 11th International Conference on Computer Vision* (2007), pp. 1–8. doi:10.1109/ICCV.2007.4409077. 2

[LK*81] LUCAS B. D., KANADE T., ET AL.: An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia. 2

[LWZ*19] LU W., WAN G., ZHOU Y., FU X., YUAN P., SONG S.: Deepicp: An end-to-end deep neural network for 3d point cloud registration. *CoRR abs/1905.04153* (2019). URL: http://arxiv.org/abs/1905.04153, arXiv:1905.04153. 2

[PRG*20] PAIS G. D., RAMALINGAM S., GOVINDU V. M., NASCIMENTO J. C., CHELLAPPA R., MIRALDO P.: 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 7193–7203. 2

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 2

[SHT09] SEGAL A., HÄHNEL D., THRUN S.: Generalized-icp. doi:10.15607/RSS.2009.V.021. 2, 4

[SWK07] SCHNABEL R., WAHL R., KLEIN R.: Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*, 2 (June 2007), 214–226. 1

[ZPK18] ZHOU Q.-Y., PARK J., KOLTUN V.: Open3D: A modern library for 3D data processing. *arXiv:1801.09847* (2018). 3, 4

[ZSN*17] ZENG A., SONG S., NIESSNER M., FISHER M., XIAO J., FUNKHOUSER T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR* (2017). 1