Point clouds alignment with incomplete shapes

Dan Halperin
Technical University of Munich
Munich, Germany
dan.halperin@tum.de

Andrii Protas
Technical University of Munich
Munich, Germany
andrey.protas@tum.de

Manuel Coutinho
Technical University of Munich
Munich, Germany
manuel.coutinho@tum.de

Priya Arumugam
Technical University of Munich
Munich, Germany
Priya.arumugam@tum.de

02 August, 2022

Abstract

Finding the affine transformation between two point clouds is an ongoing research work in computer vision and has applications in computer graphics, robotics, etc. Several powerful algorithms have been proposed for this purpose, such as the Iterative closest-points (ICP). However, it has been shown that they tend to converge to poor optima or saddle points. In this project, we investigated the use of a deep learning approach, Deep-Closest Point (DCP), to solve the problem. However, their work assumes complete shapes as source and target, while real-world applications often take incomplete shapes as an input. Therefore, we decided to address this issue. We used transfer learning on their given model, by extending the training set to handle incomplete shapes. In addition, we performed a comparison of the use of DCP and ICP.The results show that the combination of DCP and ICP is the best. While ICP succumbs to saddle points alone, DCP manages to push it pass them, like with Momentum.

1 Introduction

Point cloud alignment has been an important research topic in recent years. The goal is to match similar shapes in different scenes and shape representations to obtain better 3D reconstructions from multiple inputs. While algorithms such as Iterative Closest Point (ICP) [2] or Deep Closest Point (DCP) [9] are very successful when they receive two complete and very similar shapes as input, it turns out that they struggle with the more difficult cases where the corresponding shapes are incomplete, resulting in a larger mismatch. The problem of matching incomplete shapes better simulates real-world scenarios, such as 3D scans, which are often incomplete and do not match as expected. In this project, we experimented with different configurations and compared their effects on the performance of the DCP model.

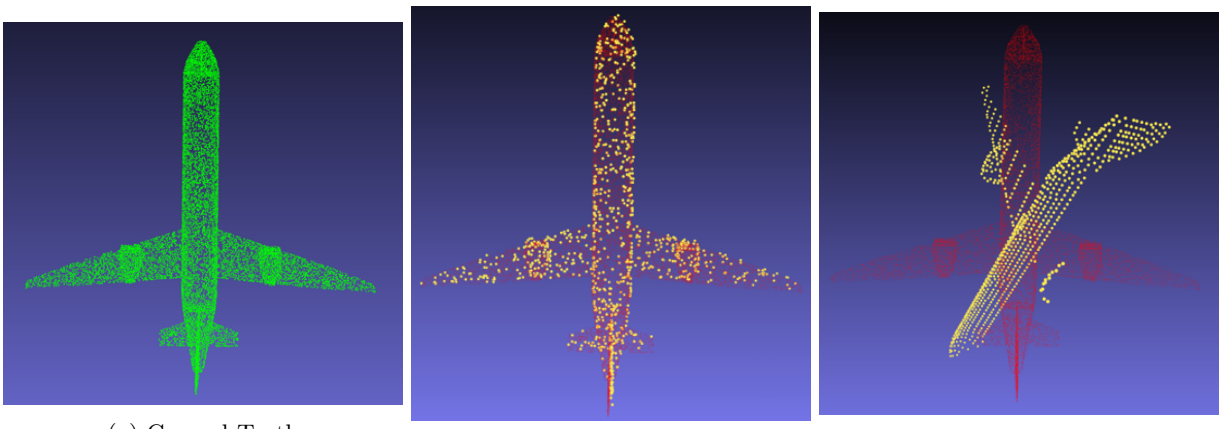
In the original proposal, we had indicated that we wanted to study the comparisons of point clouds that were artificially and randomly downsampled. However, this must already be done, and the full shapes, consisting of 16,384 points, must be downsampled to only 1024 points before being fed into the DCP backbone. Therefore, our objective was only to see if the reconstruction of partially realistically sampled shapes would improve the transformation prediction, and to experiment with the addition of the ICP algorithm.

First, we checked the performance of the original DCP model on the associated ModelNet [10] dataset. Second, we experimented with distribution shifts when applying the same model to the ShapeNet dataset. Third, we tried applying the same DCP model to match incomplete shapes and got very poor performance. Next, we used SpareNet [11], a backbone from Microsoft trained directly on the ShapeNet dataset, to reconstruct the partial shapes. When we fed these reconstructed shapes into the DCP model, we could observe significantly better results. Finally, dramatic performance improvements were obtained for partial and reconstructed shapes after retraining the DCP model with a mixture of partial and complete shapes (at the cost of a small performance loss for complete shapes).

2 Related Work

2.1 SpareNet - 3D reconstruction

3D scanning of point clouds and shapes has been widely explored and has seen an exponentially growing number of applications using LiDAR or depth cameras. However, online raw scans of point clouds are typically sparse and incomplete due to sensor viewing angles and occlusions. Following the success of StyleGAN [7], SpareNet [11] uses a generative model trained in a supervised manner to



(a) Ground Truth (b) Identical Shapes Prediction (c) Partial Shapes Prediction

Figure 1: Original DCP model's predictions on the ShapeNet dataset. In figures (b) and (c), red represents the GT target and yellow represents the prediction. Although it performs well on identical shapes, it performs poorly on partial shapes. All predictions are made with 1024 points. The full point clouds are only for visualization.

reconstruct partial shapes collected from ShapeNet [3] and KITTI [4] datasets. The authors claimed that they achieved state-of-the-art results.

The architecture consists of an embedding encoder, which is fed to a StyleGAN [7], to reconstruct style like features, which in turn are fed into an MLP. One of the novelties of this method is the double loss functions. The first is a trivial point cloud distance measurement, as the chamfer-distance [6] or Earth Mover's distance [8], while the second is and MSE, applied on heatmaps that are created out of the predicted point cloud and the ground-truth.

In this project, we used a pre-trained model of SpareNet as a basis to obtain an accurate estimate of the real shape before recovering the rotation and translation matrices R and t, respectively (Figure 2).

2.2 DCP - Deep closest points

As mentioned earlier, the Deep Closest Points (DCP) [9] is a deep learning approach to approximate the real-world transformation between two point clouds.

The architecture consists of 3 layers: An embedding layer, which takes two point clouds of 1024 points and converts them into 512 features length vector embedding. It is crucial to note, that the algorithm does not perform well with a different number of points. Then, embeddings are sent to a transformer-head, which extract the global information and the relations between the points within the clouds. In the end, the Procrustes-alignment algorithm [5] to recover the Rotation and translation matrices. This part, however, is not learnable.

In their work, the authors claimed to overcome the obstacles of convergence to a bad minima or saddle points. However, since the dataset ModelNet [10] used consists only of complete shapes, it still struggles with incomplete shapes. In our project, we approached the problem in two ways. First, we extend the training data to generalize to partial point clouds and re-train the model using transfer learning. The other way is to first feed SpareNet with the incomplete form and then perform DCP inference on the new reconstructed form.

3 Datasets

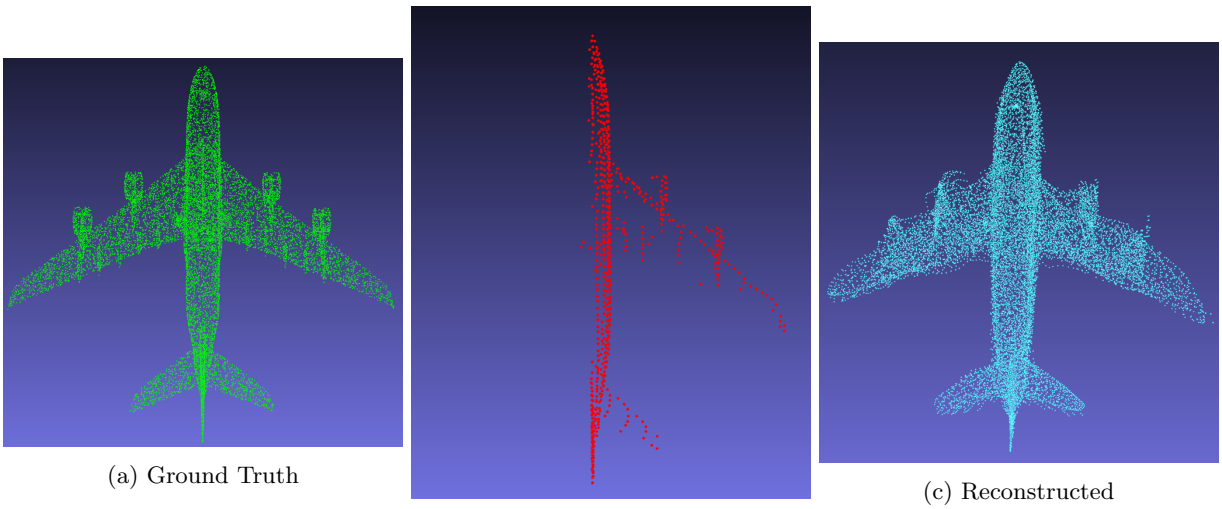
We work hands on with two different datasets.

ModelNet [10] is a clean collection of 3D CAD models for objects, consisting of 40 different categories of shapes. All CAD objects, which are annotated computer made objects, in this dataset are full. The original DCP [9] paper bases its work on top of this dataset.

ShapeNet [3], on the other hand, covers 55 common object categories with about 51,300 unique 3D models. Each model is reconstructed from 8 different views of sub-point clouds. Therefore, it is a suitable foundation for work like SpareNet to build upon. In our project, we decided to rely mainly on this dataset for rotation and reconstruction.

4 Method and Experiments

Since ModelNet consists only of scaled shapes, it was necessary to choose a different dataset that contained partial clouds. However, when testing the DCP model on ShpeNet, we found that the original model performed similarly on both the ModelNet and ShapeNet datasets, even though they were from a different distribution. However, as shown in Table 2, the tests with the partial point clouds showed poor performance, i.e., the model accurately estimated the transformation as long as the shape we were trying to align exactly matched the original shape. It is important to note, that only the rotation



(a) Ground Truth

(b) Partial Point Cloud

(c) Reconstructed

Figure 2: Reconstruction with the Style-Generative inspired SpareNet.

estimation performed poorly, as the translation in the Procrustes Alignment [5] is calculated as the difference between the point clouds means. Hence, in our visualizations, we could see that the translation is estimated accurately.

Therefore, our first experiment was quite obvious, namely to perform transfer learning by extending the training dataset to partial forms as well. Our setup was a Google Cloud virtual machine with the weak Nvidia-80. A promising idea was to merge the two backbones of SpareNet and DCP and train a model directly on the reconstructed point clouds, but without penalizing the reconstruction pipeline. In this way, rather than learning the best visual reconstruction, the network could use the power of generation to create shapes that rotate more accurately. However, this idea was eventually dropped because our Google Cloud setup was too weak for the task, both in terms of sufficient GPU memory and computational power.

When loading the data, we made a split of 40% of full shapes and 60% of partial ones, for both the training and validation datasets. However, the architecture does not allow us to freeze some learnable layers, since all parts must be modified as described in subsection 2.2 to generalize to the new shapes. We set a high dropout probability and a high weight decay coefficient for regularization, trying to force the pre-trained weights to generalize to the new data. However, the training process was stopped after one epoch by the mechanism of early stopping. Nevertheless, as can be seen in Table 1, while performance actually degraded for complete shapes, point clouds that did not follow the exact same structures as their targets saw a significant increase in performance.

In the second experiment, we wanted to test the idea of reconstructing a partial input before feeding it into the DCP pipeline. Before the training described above, the reconstruction also performed poorly because it did not fully match the structure of the ground truth. Now, after the new training, the partial form's reconstruction performed almost as well as the ground truth version, while the partial form still lagged behind despite the great improvement (about 6.5 times better).

Moreover, in their paper [9], the authors of DCP claimed that using DCP and ICP [2] together further increases the performance. Therefore, we implemented the ICP algorithm and compared the results, as can be seen in Table 1. While the combination of DCP and ICP achieved about 2.5 times better performance, it struggled with about 10 times longer runtime compared to the ICP-only algorithm.

However, it should be noted that our experiments were limited to only a small magnitude of rotation and translation, i.e., rotation angles in the range $[0, 45]$ of degrees, and the translation entries were sampled from $[-0.5, 05]$. In addition, testing with hyperparameters for the sampled random transformation that were different from those in the original repository [1] resulted in very poor performance.

5 Metrics

The transformation for each sample in the dataset was sampled randomly. Then, the DCP algorithm was tested in trial to recover it. Hence, a metric is required to evaluate how well the estimations are. The rotation matrix holds $R \in SO(3)$, and therefore is orthogonal, thus $R^T = R^{-1}$. Therefore, we use the **Mean Squared Error** loss function for the recovery of R_{pred}, to estimate how different it is from the inverse matrix of the original rotation matrix R_{orig}:

		DCP		DCP + ICP		ICP	
Point Cloud	**Model**	Error	Runtime (S)	Error	Runtime (S)	Error	Runtime (S)
Full	Original	$2.79e-06$	0.07	$4.21e-15$	0.12	$1.21e-04$	0.04
Full	Retrained	$2.40e-04$	0.07	$1.63e-05$	0.11	-	-
Partial	Original	$1.34e-02$	0.08	$1.33e-02$	0.11	$9.25e-04$	0.03
Partial	Retrained	$5.22e-04$	0.08	$4.15e-04$	0.10	-	-
Reconstructed	Original	$1.26e-02$	0.588	$1.30e-02$	0.511	$8.73e-05$	0.056
Reconstructed	Retrained	$2.32e-04$	$5.94e-05$	$1.20e-05$	0.576	-	-

Table 1: Comparison of the ICP and DCP on the ShapeNet [3] dataset, both of the error values and the runtimes. Scores were evaluated as a mean over 50 samples, while runtime was evaluated on a single input.

Dataset	**model**	**Full**	**Partial**	**Reconstructed**
ModelNet [10]	Original	$1.09e-05$	-	-
ShapeNet [3]	Original	$2.79e-06$	1.34e-02	1.26e-02

Table 2: Comparisons between the ModelNet and ShapeNet datasets. Although the model was originally trained on ModelNet, it performs very well right away for identical shapes taken from ShapeNet. However, non-identical shapes perform poorly.

$$MSE(R_{pred}, R_{orig}) = \frac{1}{3 \times 3} ||R_{pred}^T \cdot R_{orig} - I(3)||_2^2$$

without a need to compute the inverse matrix directly, and where I(3) is the identity matrix of size 3. The evaluation for the translation is also done by the *MSE* loss metric, as it is the metric according to which the model is trained [9].

6 Results

In this section, we would like to discuss the results we collected in the Table 1 and Table 2 tables.

First, we have achieved the goal of the project. We succeeded in showing that a pipeline fed with a partial point cloud can leverage the powerful capabilities of 3D reconstruction models to recover a better transformation between a source and a target. Our generalized model has shown increase in improvement in transforming non-identical shapes, such as partial and reconstruction, at the price of lower performance than the original model for identical shapes. Moreover, the most interesting trend of Table 1 is the performance of the trained model in combination with the ICP. We observed that the ICP alone, implemented with a simple distance metric, converged to poor minima, as described in the original paper, with or without the application of the DCP before. However, the newly trained DCP allowed the ICP to converge to lower error rates when used together, which strengthens our confidence that a further training and fine-tuning will produce intriguing results.

7 Conclusion

In summary, we have shown in this project that reconstructing a partial point cloud before restoring the transformation to a target cloud is beneficial and promising. We explain the immediate low error rate for full clouds from the ShapeNet dataset, even though it is not from the same distribution as ModelNet, on the grounds that the DCP model generalizes to the easy case of finding correspondences between two clouds, when both are identical. Moreover, our short transfer learning training has already shown the potential, although the sampling space for the transformations was quite limited. Despite the claim that ICP could get stuck in saddle points, DCP helped it avoid them and converge to promisingly low error rates, when the ICP was applied after a well performed transformation by the DCP beforehand. That said, while the reconstruction pipeline with DCP+ICP extension has shown the best results, the version without the reconstruction stage showed competitive results, with 5 times less runtime. However, a great deal of generalization is yet to be made.

The most interesting direction for future work would be to set up a more powerful system and train the model in an end-to-end fashion, so that the SpareNet architec-

ture should not learn to reconstruct for minimal disparity from a ground truth cloud, but find the best reconstruction, as an auxiliary tool for the transformation recovery pipeline. In addition, we believe that using the powerful combination of DCP and ICP, alongside a trained model on a wider dataset is correct, as ICP provides a large performance boost with negligible runtime overhead, and eventually is the best overall performing model.

References

[1] Dcp github repository. https://github.com/WangYueFt/dcp. Repo of DCP. 3

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. 1, 3

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. 2, 4

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2

[5] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):285–321, 1991. 2, 3

[6] Andras Hajdu, Lajos Hajdu, and Robert Tijdeman. Approximations of the euclidean distance by chamfer distances, 2012. 2

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018. 1, 2

[8] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. 2

[9] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019. 1, 2, 3, 4

[10] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 1, 2, 4

[11] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion, 2021. 1