

# COMPLETION OF SPARSE AND PARTIAL POINT CLOUDS OF VEHICLES USING A NOVEL END-TO-END NETWORK

Yan Xia<sup>1,\*</sup>, Weiyan Liu<sup>2</sup>, Zhipeng Luo<sup>2</sup>, Yusheng Xu<sup>1</sup>, Uwe Stilla<sup>1</sup>

<sup>1</sup> Photogrammetry and Remote Sensing, Technical University of Munich, 80333 Munich, Germany  
(yan.xia, yusheng.xu, stilla)@tum.de

<sup>2</sup> Fujian Key Laboratory of Sensing and Computing, School of Informatics, Xiamen University, 361005 Xiamen, China  
wqliu1026@163.com, zpluo@stu.xmu.edu.cn

## Commission II, ICWG II/III

**KEY WORDS:** Shape Completion, Uniform Point Cloud, Point Cloud Generation, 3D Reconstruction, Deep Learning

### ABSTRACT:

Completing the 3D shape of vehicles from real scan data, which aims to estimate the complete geometry of vehicles from partial inputs, acts as a role in the field of remote sensing and autonomous driving. With the recent popularity of deep learning, plenty of data-driven methods have been proposed. However, most of them usually require additional information as prior knowledge for the input, for example, semantic labels and symmetry assumptions. In this paper, we design a novel and end-to-end network, termed as S2U-Net, to achieve the completion of 3D shapes of vehicles from the partial and sparse point clouds. Our network includes two modules of the encoder and the generator. The encoder is designed to extract the global feature of the incomplete and sparse point cloud while the generator is designed to produce fine-grained and dense completion. Specially, we adopt an upsampling strategy to output a more uniform point cloud. Experimental results in the KITTI dataset illustrate our method achieves better performance than the state-of-arts in terms of distribution uniformity and completion quality. Specifically, we improve the translation accuracy by 50.8% and rotation accuracy by 40.6% evaluating completed results with a point cloud registration task.

## 1. INTRODUCTION

Recently, 3D shape completion for vehicles from real scanned data is regarded as a fundamental and intriguing area in the fields of remote sensing, computer vision, and robotics. Especially, with the development of the autonomous driving field, more and more companies and research institutes adopt different sensors (i.e., mobile laser scanning systems or depth cameras) to collect 3D point clouds in urban scenarios. However, acquired point clouds of the real world from these sensors are usually incomplete and noisy. For example, as shown in Figure 1 (Top), illustrated point clouds of cars from the KITTI dataset (Geiger et al., 2013) are sparse and incomplete, even some of them have a very few numbers of points, which is counter-productive to the further recognition task. To tackle this problem, the completion of low-quality point clouds via recovering geometric surfaces from sparse and partial points could be a practical but challenging solution. In case the point clouds of vehicles can be completed, it would be beneficial for the recognition algorithm in the autonomous driving field and provides a solid foundation for the perception tasks (Wen et al., 2019).

Giving a partial and sparse point cloud of vehicles, the shape completion aims to recover the missing points and output a uniform and dense point cloud with complete geometry. With the recent popularity of deep learning-based methods and the emergence of the synthetic datasets of 3D CAD models, such as Pix3D (Sun et al., 2018), ObjectNet3D (Xiang et al., 2016), and ShapeNet (Chang et al., 2015), using a data-driven strategy with neural networks to accomplish this mission has proved to be feasible and realistic. For example, network structures, like 3D-EPN (Dai et al., 2017) and SSCNet (Song et al., 2017), have revealed the excellent performance of inferring 3D shape

information through leaning on the synthetic datasets. However, these methods usually adopt a voxelization representation to organize dataset, which limits the output resolution since 3D voxel grids are actually a down-sampling. To some extent, these voxel-based methods are difficult to achieve fine-grained completions as well, because they are an artifact of artificial discretization.

To this end, more and more researchers choose to directly operate on raw point clouds. Compared with 3D voxelization representation, point clouds based representation can boost the performance of completion in terms of fine-grained details of objects. Recently, the Point Completion Network (PCN) (Yuan et al., 2018) was proposed to produce a dense and complete geometric shape from a partial 3D point cloud. However, points in its output are particularly non-uniform, which often over-concentrated in areas like the wheels of vehicles. Inspired by this work, in this paper, we propose an innovative 3D shape completion network, termed S2U-Net, and the main contributions of this paper are as follows:

- We design a novel network S2U-Net, which operates on a partial point cloud directly for 3D shape completion. By applying an upsampling strategy, S2U-Net can recover a uniform, dense, and complete point cloud for vehicles from real scan data.
- Experimental results show the improved performance of the S2U-Net over the state-of-arts, and the completed results can significantly help for the point cloud registration task.

We train the proposed S2U-Net on the class of vehicles from the synthetic dataset ShapeNet (Chang et al., 2015) and test it

\* Corresponding author

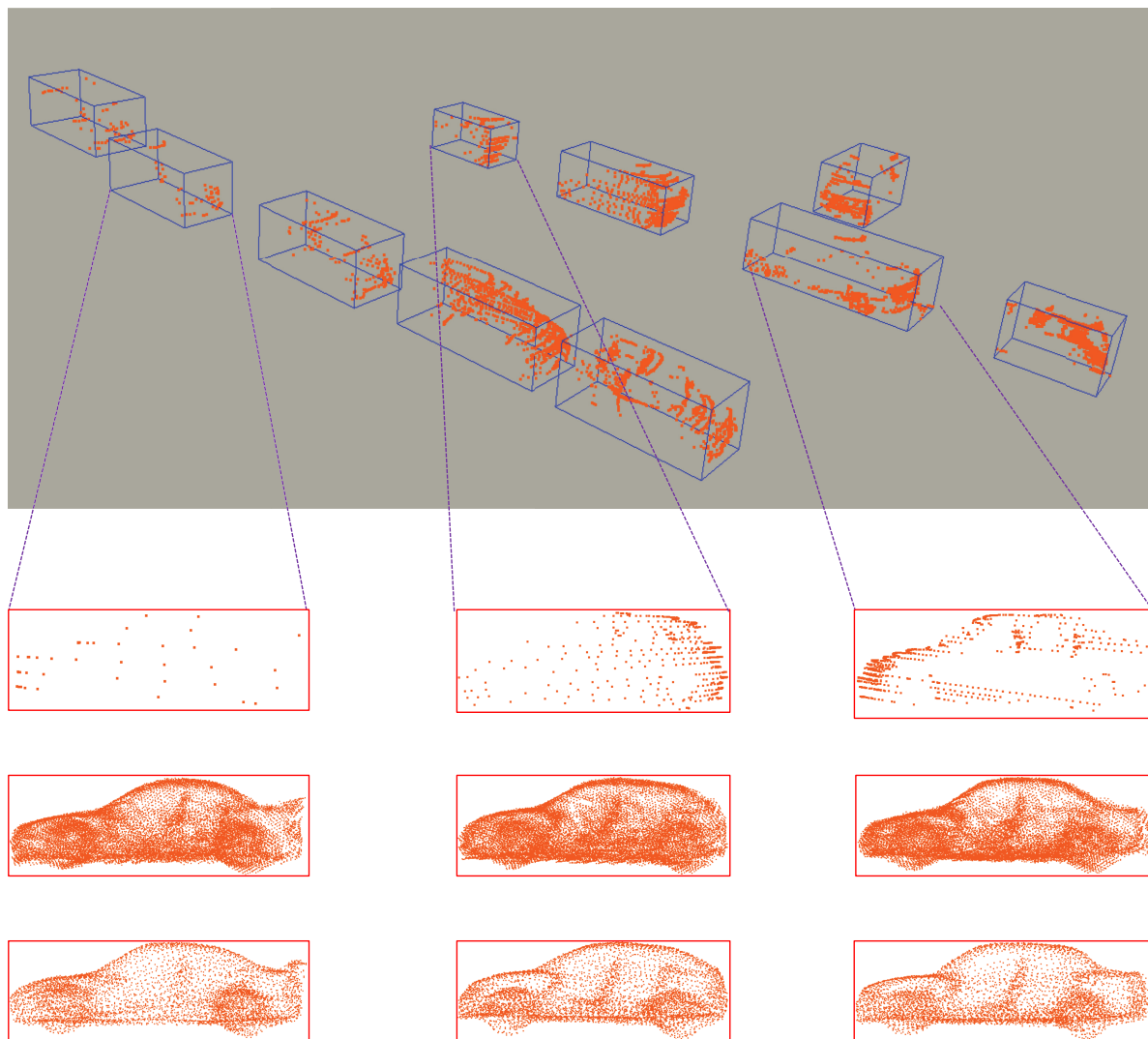


Figure 1. (Top) Real scan data collected from KITTI (Geiger et al., 2013) dataset. (Middle) Completion results generated by PCN (Yuan et al., 2018). (Bottom) Completion results by our proposed S2U-Net.

on two real-scanned datasets KITTI (Geiger et al., 2013) and TUM dataset (Gehring et al., 2017). Furthermore, we also investigate the completed performance based on the results of point cloud registration. Experimental results demonstrate that our method has a better performance than the state-of-arts in respect of distribution uniformity and complete quality.

## 2. PREVIOUS WORK

### 2.1 Geometry-based Shape Completion

According to geometric cues (e.g., continuity of local surfaces or volumetric smoothness) of incomplete inputs, geometry-based approaches (Kazhdan et al., 2006, Tagliasacchi et al., 2011, Wu et al., 2015a) can successfully retouch small holes on surfaces of point clouds. When recovering significant missing regions, hand-designed heuristics are applied to complete the 3D shape of objects. For example, (Schnabel et al., 2009) employed a series combination of planes and cylinders to guide the 3D shape completion for the partial point cloud. Furthermore,

(Li et al., 2011) proposed an innovative method to learn global relations between these primitives. Considering that man-made objects usually have structural regularity, some studies (Pauly et al., 2008, Zheng et al., 2010) proposed approaches to find regular or periodic structures in geometric models and then use them to complete missing surfaces. However, these methods heavily rely on the assumption that the partial point cloud as input is of moderate degrees of completion already.

### 2.2 Template-based Shape Completion

Apart from the geometry-based completion, another common shape completion strategy is to retrieve the most similar shape or templates from a large-scale database as a reference, then to deform or reconstruct the input shape according to the retrieved reference. (Pauly et al., 2005) produced a complete 3D shape using geometric priors for missing regions from a given 3D shape database, but it requires manual interaction to limit categories of objects. Similarly, (Rock et al., 2015) explored a method to complete a 3D model of any class automatically from one depth image. However, these methods strongly depend on

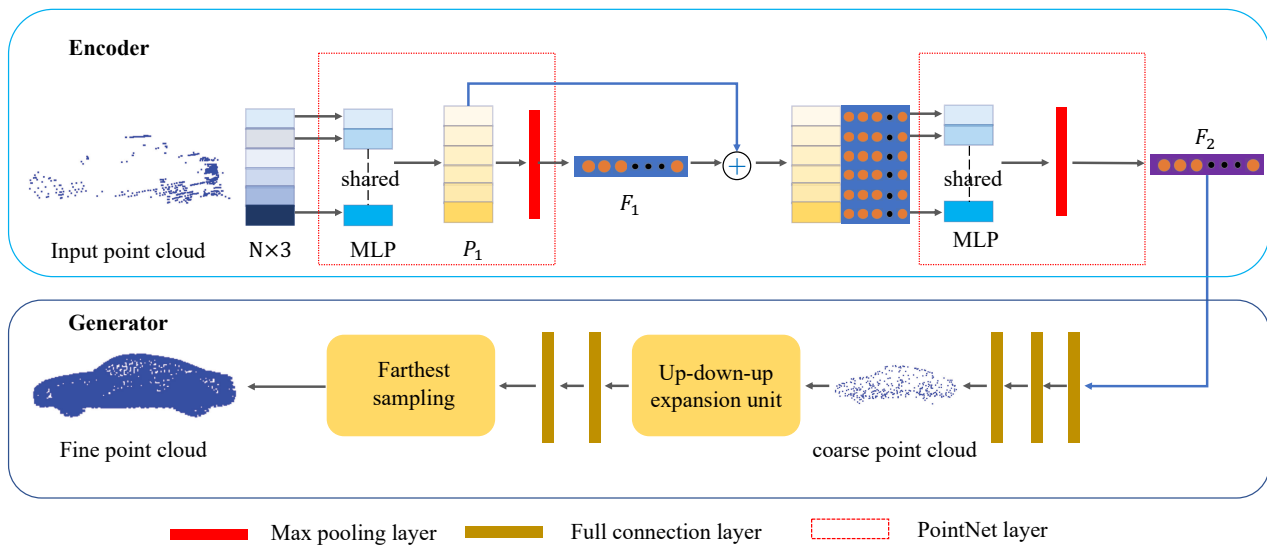


Figure 2. A pipeline of the proposed S2U-Net, in which the global feature is extracted by an encoder network. Then, a coarse-to-fine strategy is applied for the following generator.

the capacity of the 3D shape database. To avoid the high dependency of large databases, (Shen et al., 2012) conducted an assembly approach of geometric primitives to recover 3D structures with a small-scale shape dataset. (Sung et al., 2015) applied a method to predict the geometric information of an input model and then used a global optimization to reconstruct the entire underlying surface. However, these methods still suffer from several limitations. Firstly, the optimization schemes are usually too expensive in computational cost for the online application. Secondly, each shape in the pre-prepared 3D shape database requires to be labeled and segmented manually. Last but not least, they are always sensitive to noise.

### 2.3 Learning-based Shape Completion

Recently, deep learning-based methods for 3D shape completion has become a popular topic. Most of them provided outputs completing shapes directly from a partial input using an end-to-end artificial neural network. (Wu et al., 2015b) constructed a large-scale synthetic object dataset named ModelNet and proposed a Convolutional Deep Belief Networks (CDBNs) to learn shape distributions for completing point clouds. (Thanh Nguyen et al., 2016) integrated CDBNs and Markov Random Fields (MRFs) to recover incomplete shapes. (Sharma et al., 2016, Varley et al., 2017) implied an Encoder-Decoder Network for shape completion. However, these methods all selected voxelization as 3D data representation since it can be applied in the 3D convolution. (Dai et al., 2017) explored a 3D Encoder-Predictor Network (EPN) for estimating a sparse but complete shape, then refined this shape through the nearest-neighbor-based volumetric post-processing. One recent work, (Yuan et al., 2018) proposed to directly operate on point clouds for 3D shape recovery, which is most related to our demand. Nevertheless, this manner exits two limitations. One is that the shape completed using this approach is not uniform, with most of the regions are over-concentrated. While another is that output point clouds always lose some detailed information.

## 3. METHOD

### 3.1 Overview

We convert the completion of point clouds into a set problem: given a partial sparse point cloud set  $X$  obtained from the observed surfaces, we aim to predict a dense and uniform point cloud set  $Y$ , which uniformly sampled from the complete surfaces of the object. Notably,  $Y$  is not necessarily a superset of  $X$ . They are sampled from the object surfaces independently.

Our proposed S2U-Net consists of two modules of the encoder and the generator. The former is designed for extracting the global feature map of the partial input, while the latter one contributes to producing a dense, complete, and uniform point cloud from the latent space. Besides, we utilize a coarse-to-fine training approach in the generator: firstly predicts a sparse point cloud with complete geometry, then refines it with local regions using an upsampling procedure. For a better understanding, the network architecture of the S2U-Net is shown in Figure 2, which includes two components: (1) To robustly extract the global features of the partial inputs, two stacked PointNet (Qi et al., 2017a) layers are employed in the encoder. (2) To generate the dense, uniform, and complete point cloud, the S2U-Net network uses the coarse-to-fine strategy in the generator. The coarse network estimates the complete but sparse point cloud. Then, the fine network upsamples the sparse point cloud by combining local information and global structure. The detailed information of network architecture is introduced as the following sub-sections.

### 3.2 Encoder

Our encoder is based on the recently advanced feature extraction network PointNet (Qi et al., 2017a), which directly operates on the point clouds. Besides, its extension PointNet++ (Qi et al., 2017b) applied the PointNet layers as a local feature extraction unit to learn point cloud features more efficient and robust. Inspired by this, the encoder, as illustrated in top of Figure 2, adopts two stacked PointNet layers to extract the geometric information for the input point cloud. Each PointNet layer

comprises of one shared multiple layer perception (MLP) and one max-pooling layer as a basic module. In the first PointNet layer, we learn a point-wise feature  $P_1$  from  $N$  input points of  $N \times 3$  through MLP, where 3 is the three dimensions coordinates of each point. Afterwards, a max-pooling layer is employed on  $P_1$  to output a 1024-dimensions global feature vector  $F_1$ . In the second PointNet layer, we firstly concatenate the global latent space with per independent point feature by feeding  $F_1$  back to the point-wise features  $P_1$ . The final global feature  $F_2$  is extracted build with the aggregated point features.

### 3.3 Coarse-to-fine Generator

After obtaining the final global feature  $F_2$ , the generator, as shown on the bottom of Figure 2, aims to predict 3D coordinates of the output point cloud. We explore a coarse-to-fine training approach for generating the complete point cloud. Firstly, inspired by 3D object reconstruction network RealPoint3D (Xia et al., 2019), fully-connected layers work well for estimating a sparse set of points which have the complete geometric information of the object. Therefore, the feature  $F_2$  is fed to the following three fully connected layers in the shape of 3072, and then we reshape the vector to a coarse point cloud with a size of  $1024 \times 3$ .

However, since the regression of points using fully connected layers is not restrictive on the local density, a large number of points will be over-concentrated when we only employing fully connected layers to output the dense point cloud. Thus, in the second step, we apply a feature expansion strategy (Li et al., 2019) to enhance the distribution uniformity and completeness of output point cloud, where an up-down-up expansion unit is used. The up-down-up expansion unit consists of the up-feature operator and down-feature operator, which have been proven to enhance the feature variations effectively. Last, we add the farthest sampling step to retain 16384 points in the output point cloud.

### 3.4 Loss Function

For the generated result and the ground truth, we expect their topological distance to be the smallest. The distance function must be highly efficient and invariant to permutations of the points. Furthermore, it needs to be robust to outliers in the sets because each point cloud is all unordered point sets with underlying noise. Inspired by (Fan et al., 2017), we explore Chamfer distance (CD) and Earth Mover's Distance (EMD) to optimize the S2U-Net network.

CD is defined as follows:

$$CD(Q_1, Q_2) = \sum_{p \in Q_1} \min_{q \in Q_2} \|p - q\|_2 + \sum_{p \in Q_2} \min_{q \in Q_1} \|p - q\|_2 \quad (1)$$

where  $Q_1, Q_2 \subseteq R^3$ ,  $Q_1$  and  $Q_2$  are the generated point cloud and the ground truth, respectively. Notably, the size of  $Q_1$  and  $Q_2$  can be different.

EMD is defined as follows:

$$EMD(Q_1, Q_2) = \min_{\phi: Q_1 \rightarrow Q_2} \sum_{p \in Q_1} \|p - \phi(p)\|_2 \quad (2)$$

where  $Q_1, Q_2 \subseteq R^3$ ,  $\phi: Q_1 \rightarrow Q_2$  is a bijection. Unlike CD, the size of  $Q_1$  and  $Q_2$  must be same.

In this work, the loss function involved with two terms since we imply a coarse-to-fine training method, which introduced by as follows:

$$L(P_{coarse}, P_{fine}, P_{gt}) = EMD(P_{coarse}, \tilde{P}_{gt}) + \gamma CD(P_{fine}, P_{gt}) \quad (3)$$

where  $P_{coarse}$  is the sparse point clouds generated by the first step in the generator,  $\tilde{P}_{gt}$  is the sub-sampled ground truth point cloud with the equal size as  $P_{coarse}$ . The second term is the Chamfer distance between the final predicted point cloud  $P_{fine}$  and the full ground truth point cloud  $P_{gt}$ , in which  $\gamma$  is a hyper parameter to balance the relationship of them.

## 4. EXPERIMENTS

For evaluating the effectiveness of our S2U-Net, experiments related to 3D shape completion are conducted on synthetic shapes from the ShapeNet (Chang et al., 2015) dataset and real raw point clouds from KITTI (Geiger et al., 2013) dataset and TUM dataset (Gehring et al., 2017). We firstly introduce how to prepare the training data and some implementation details about our network. Next, we compare the proposed method with the state-of-arts. Last, we apply the completed results to point cloud registration task in order to demonstrate the performance of our method.

### 4.1 Dataset and Implementation Details

**4.1.1 Dataset** We train our network based on the ShapeNet (Chang et al., 2015) dataset, which is a richly-annotated and large-scale source of synthetic 3D CAD models. ShapeNet has covered 220,000 models and 3,135 categories of objects. In this work, we only use the category of cars as training samples here, having a total of 5677 objects. For the preparation, on the one hand, 16384 points are sampled uniformly from the mesh surfaces of each object as the ground truth point cloud. On the other hand, we create partial point clouds by transforming the depth images to point clouds. For the data augmentation, multiple depth images are used from different viewpoints (e.g., eight directions) for each sample. Therefore, each training set includes eight partial point clouds and one corresponding ground truth point cloud. Notably, the size of these partial point clouds can be different.

In addition, we choose the KITTI (Geiger et al., 2013) dataset for evaluating the performance of our method on real scan data. KITTI dataset provides raw point clouds collected by Velodyne HDL-64E rotating 3D laser scanner and annotations for the seven classes of the dynamic objects in the form of 3D bounding box tracklets. In this work, we select the category of cars for shape completion.

**4.1.2 Training strategy** The proposed network S2U-Net is implemented in the Tensorflow framework and trained on a single Nvidia Titan Xp GPU. We use Adam as an optimizer in the whole network for 100 epochs. Due to the limitation of GPU resources, the batch size is set as 8. The coarse output point cloud in the generator includes 1024 points. We set the initial learning rate as 0.0001, and it is gradually reduced by a decay rate of 0.7 per 50K steps until  $10^{-6}$ .

### 4.2 Results and Comparisons

**4.2.1 Shape completion on Real-scanned data** In this experiment, we compare our proposed method S2U-Net with

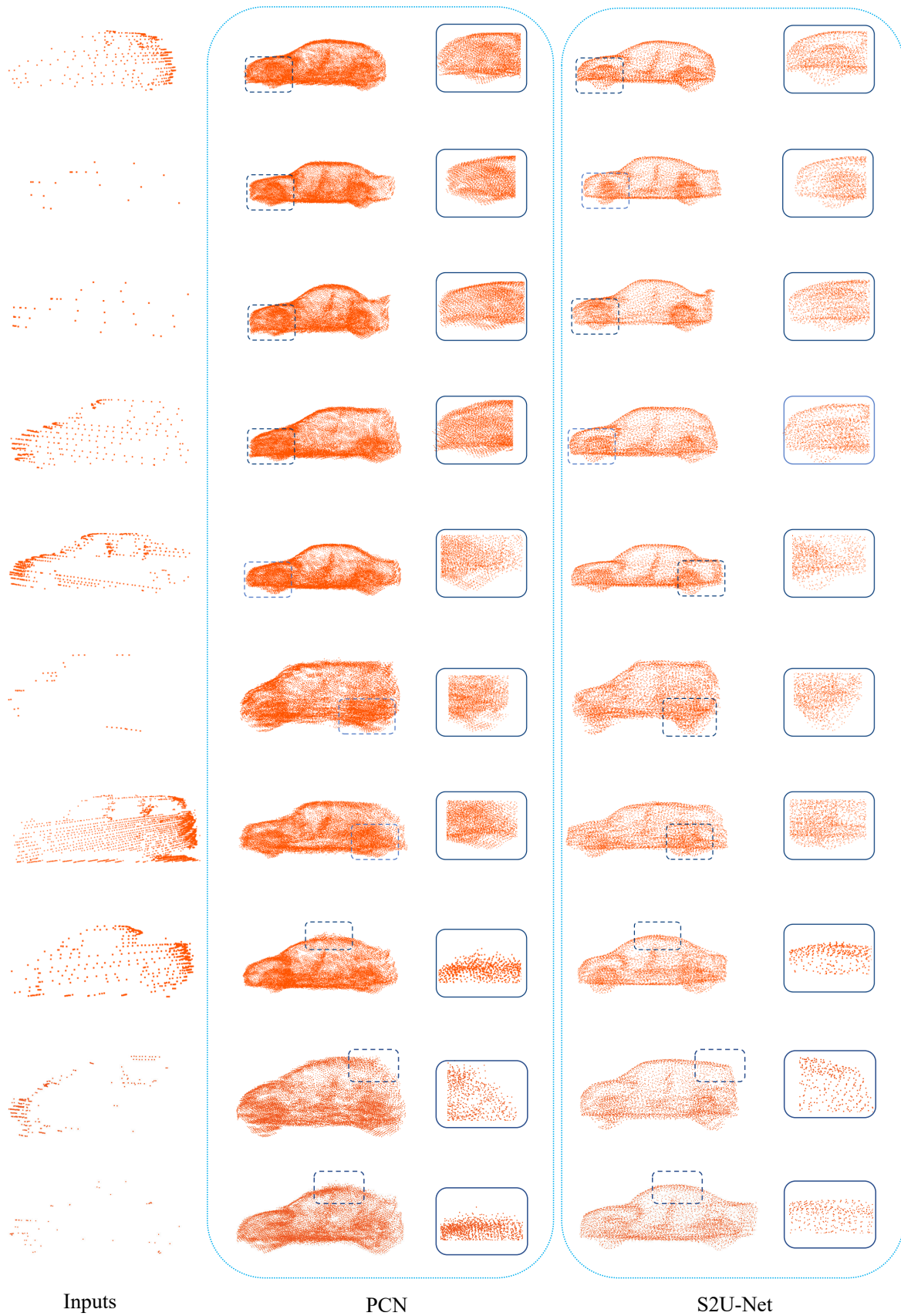


Figure 3. 3D point cloud completion for the shape of cars from the KITTI dataset. The orders from left to right: input partial scans, PCN and S2U-Net. Note that these completed point clouds all have 16,384 points.

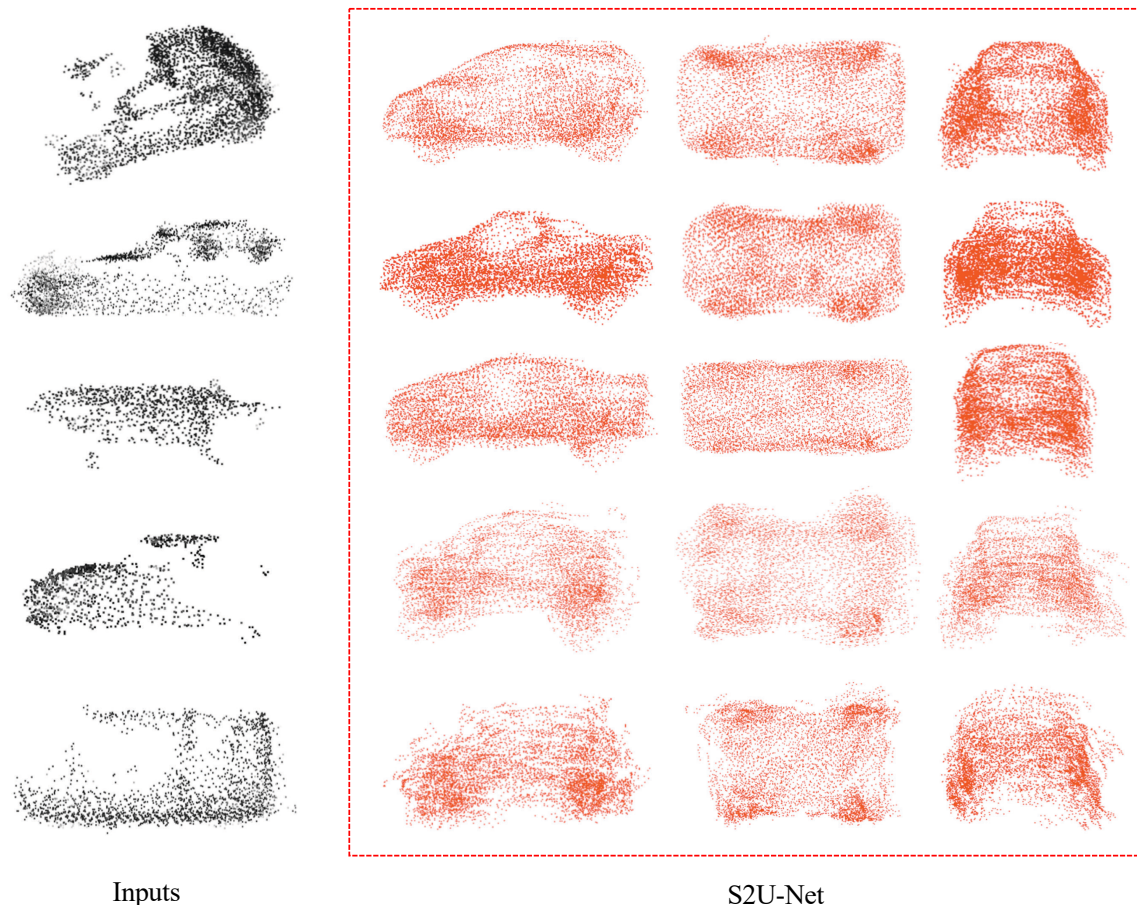


Figure 4. 3D point cloud completion for the shape of cars from the TUM City Campus dataset. The orders from left to right: input partial scans, completed results by S2U-Net.

PCN(Yuan et al., 2018) on partial real-scanned data from KITTI (Geiger et al., 2013) dataset. Following the experimental setting in PCN, we extract 2483 partial point clouds of cars according to their labeled bounding boxes for each frame. Note that we need to convert the origin scan data firstly into the box's coordinates since the training model based on cars in the synthetic 3D model dataset, in which all point clouds are normalized. Besides, because there are no ground truth point clouds of vehicles in the KITTI dataset, we only qualitatively compare these two methods. Some visual results are shown in Figure 3.

To conclusively prove the better performance of our proposed method, we randomly select ten different cars from the KITTI dataset for comparison. The first column is the sparse and incomplete input; the second and fourth columns are the results of PCN and S2U-Net, respectively. Note that the points of output point clouds are all 16384. It may lead to a visual illusion that PCN have more points since the generated point clouds by PCN are clustered and disorganized. From Figure 3, we can see that our method S2U-Net produces more uniform point sets compared with PCN, which tend to generate noisy and non-uniform point clouds. For example, in the first row of Figure 3, the points of the car created by PCN are messy, and most of the points are over-concentrated in the wheels of the vehicle. However, the points of completed results by S2U-Net are uniformly scattered on the geometric surface.

Especially, from the below-up views (the third column and fifth column), it can be seen that S2U-Net recovered the fine-detailed

structure of cars. For instance, in the first row of Figure 3, S2U-Net heals the complete wheels of the vehicle, while PCN almost misses them. And in the eighth row, some points escaped the surface of the car using the PCN method, whereas, these points lie flat and uniform on the geometric plane using our method. It is attributed to the generator's feature expansion capability and farthest point sampling strategy.

In addition, our approach reveals robustness to varying input densities. As we can see in Figure 3, the points of ten inputs range from tens to hundreds. For example, in the second, third, sixth and tenth rows, although the inputs are less than 50 points, our proposed method S2U-Net still generates complete, smooth and uniform surface for the missing structure. It also demonstrates that our method has an ability in filling large gaps in point clouds. We believe that it is owing to the powerful global feature extraction capabilities of the encoder.

**4.2.2 Applications with completion results** 3D point cloud completion of vehicles is helpful for many applications in the autonomous driving field, for example, the registration between point clouds. Therefore, we conduct a set of registration experiments, which is between vehicle point clouds of adjacent frames from the KITTI dataset. As we know, it exists some errors when estimating the spatial transformation that aligns two different point clouds. To demonstrate our completion results can significantly reduce errors for point cloud registration tasks, we take the origin partial point cloud and completion results by our proposed S2U-Net as inputs, respectively.

Besides, we implement it by a point-to-point ICP (Besl, McKay, 1992) method. ICP is a classic point cloud matching algorithm, which registers the point clouds by iteratively minimizing distances between points from two scans. It is simple and easy to implement using the PCL library (Rusu, Cousins, 2011). Notably, the ICP algorithm is not the only choice for such experiments.

Inputs	Average error	
	translation (cm)	rotation (°)
Origin	13.9413	7.0546
Completed by S2U-Net	6.8581	4.1974

Table 1. The errors of point cloud registration using different inputs

Similar to PCN (Yuan et al., 2018), we select the average rotation and translation error as an evaluation metric, which is defined as following respectively:

$$E_R = 2 \cos^{-1}(2 < R_1, R_2 >^2 - 1) \quad (4)$$

$$E_T = \|T_1 - T_2\|_2 \quad (5)$$

where  $R_1$  and  $T_1$  are the rotation, translation of ground truth in the KITTI dataset, respectively.  $R_2$  and  $T_2$  are the rotation, translation measured by the ICP method, respectively.

The results are shown in Table 1. We refer to the sparse and partial point clouds from the original scans as "Origin". From Table 1, we can clearly see that using the completed point cloud by S2U-Net significantly improves the effectiveness of the registration task. In terms of translation accuracy and rotation accuracy, we promote that by 50.8% and 40.6%, respectively. We think the completed point clouds can achieve satisfying performance is due to that they have more overlapping regions from the points generated by S2U-Net than that of origin scan data. Therefore, it also demonstrates our method has a great significance for real applications.

**4.2.3 Test with TUM dataset** To demonstrate the effectiveness of our proposed method S2U-Net on real scan data, we select the Mobile Laser Scanning (MLS) dataset (Gehring et al., 2017) in the TUM City Campus for further evaluation. This dataset covers around 80000  $m^2$  with annotations, and Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) originally acquires this dataset with two Velodyne HDL-64E laser scanner. Here, we use the category of vehicles as the testing data. Some results include the failure cases that are visualized in Figure 4. Notably, there are also no corresponding real point clouds of vehicles in this dataset.

As we can see in the first three rows in Figure 4, S2U-Net produces the dense, uniform, and complete point clouds for these partial inputs with different sizes. However, the remaining rows show examples of failure cases of our proposed method. We can clearly see the generated points are not lying on the surface, with many points escaped. Because there are not bounding boxes of vehicles from the TUM City Campus dataset, we need to extract the point clouds of vehicles manually. In addition, it exists many noises in this dataset, for example, in the second row, there are many unrelated points on the wheels of the cars. Therefore, these reasons result in the accuracy of data preprocessing is not enough.

## 5. CONCLUSION

In this paper, we proposed a novel end-to-end network S2U-Net, which can recover a more uniform and fine-detailed structure for 3D shape completion of vehicles from a partial point cloud in a real scan data. Compared with previous methods, we explored an upsampling strategy to enhance the distribution uniformity and completion quality in the process of generating complete point clouds. Experimental results demonstrated the performance of our method can outperform other state-of-art methods.

However, we found the variation of generated point clouds is less noticeable. One of the reasons is that there is a little inter-class variation for vehicles in the ShapeNet dataset when we prepare training data. Thus, we believe that it is also a limitation for any method using this synthetic 3D CAD model database as the training data, including 3D-EPN, PCN, and so on. Besides, S2U-Net can only complete the class of vehicles from real scan data. We will explore the more categories in the urban scene in the future, such as traffic signs, buildings, and so on.

## ACKNOWLEDGEMENTS

This research is supported by the China Scholarship Council.

## REFERENCES

- Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. *Sensor fusion IV: control paradigms and data structures*, 1611, International Society for Optics and Photonics, 586–606.
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H. et al., 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Dai, A., Ruizhongtai Qi, C., Nießner, M., 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5868–5877.
- Fan, H., Su, H., Guibas, L. J., 2017. A point set generation network for 3d object reconstruction from a single image. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 605–613.
- Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. AN APPROACH TO EXTRACT MOVING OBJECTS FROM MLS DATA USING A VOLUMETRIC BACKGROUND REPRESENTATION. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.
- Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), 1231–1237.
- Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. *Proceedings of the fourth Eurographics symposium on Geometry processing*, 7.
- Li, R., Li, X., Fu, C.-W., Cohen-Or, D., Heng, P.-A., 2019. Pagan: a point cloud upsampling adversarial network. *Proceedings of the IEEE International Conference on Computer Vision*, 7203–7212.

- Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., Mitra, N. J., 2011. Globfit: Consistently fitting primitives by discovering global relations. *ACM SIGGRAPH 2011 papers*, 1–12.
- Pauly, M., Mitra, N. J., Giesen, J., Gross, M. H., Guibas, L. J., 2005. Example-based 3d scan completion. *Symposium on Geometry Processing*, 23–32.
- Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., Guibas, L. J., 2008. Discovering structural regularity in 3d geometry. *ACM SIGGRAPH 2008 papers*, 1–11.
- Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.
- Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 5099–5108.
- Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D., 2015. Completing 3d object shape from one depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2484–2493.
- Rusu, R. B., Cousins, S., 2011. 3d is here: Point cloud library (pcl). *2011 IEEE international conference on robotics and automation*, IEEE, 1–4.
- Schnabel, R., Degener, P., Klein, R., 2009. Completion and reconstruction with primitive shapes. *Computer Graphics Forum*, 28number 2, Wiley Online Library, 503–512.
- Sharma, A., Grau, O., Fritz, M., 2016. Vconv-dae: Deep volumetric shape learning without object labels. *European Conference on Computer Vision*, Springer, 236–250.
- Shen, C.-H., Fu, H., Chen, K., Hu, S.-M., 2012. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6), 1–11.
- Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., Funkhouser, T., 2017. Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1746–1754.
- Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., Freeman, W. T., 2018. Pix3d: Dataset and methods for single-image 3d shape modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2974–2983.
- Sung, M., Kim, V. G., Angst, R., Guibas, L., 2015. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6), 1–11.
- Tagliasacchi, A., Olson, M., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. Vase: Volume-aware surface evolution for surface reconstruction from incomplete point clouds. *Computer Graphics Forum*, 30number 5, Wiley Online Library, 1563–1571.
- Thanh Nguyen, D., Hua, B.-S., Tran, K., Pham, Q.-H., Yeung, S.-K., 2016. A field model for repairing 3d shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5676–5684.
- Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P., 2017. Shape completion enabled robotic grasping. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2442–2447.
- Wen, C., Dai, Y., Xia, Y., Lian, Y., Tan, J., Wang, C., Li, J., 2019. Toward Efficient 3-D Colored Mapping in GPS-/GNSS-Denied Environments. *IEEE Geoscience and Remote Sensing Letters*, 17(1), 147–151.
- Wu, S., Huang, H., Gong, M., Zwicker, M., Cohen-Or, D., 2015a. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6), 1–13.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015b. 3d shapenets: A deep representation for volumetric shapes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.
- Xia, Y., Wang, C., Xu, Y., Zang, Y., Liu, W., Li, J., Stilla, U., 2019. RealPoint3D: Generating 3D Point Clouds from a Single Image of Complex Scenarios. *Remote Sensing*, 11(22), 2644.
- Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S., 2016. Objectnet3d: A large scale database for 3d object recognition. *European Conference on Computer Vision*, Springer, 160–176.
- Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M., 2018. Pcn: Point completion network. *2018 International Conference on 3D Vision (3DV)*, IEEE, 728–737.
- Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N. J., Cohen-Or, D., Chen, B., 2010. Non-local scan consolidation for 3D urban scenes. *ACM Trans. Graph.*, 29(4), 94–1.