

RESEARCH

Open Access



Enhancing the performance of point cloud completion assisted with tactile information

Ming Lu^{1†}, Jian-Han Shi^{1†}, Jian Li¹, Hao-Ran Shen¹, Huai-Yu Wang¹, Chi Zhang¹, Pu-Zheng Wang¹ and Qin Wang^{1*} 

[†]Ming Lu and Jian-Han Shi have contributed equally to this work.

*Correspondence:
qinw@njupt.edu.cn

¹ Institute of Quantum Information and Technology, Nanjing University of Posts and Telecommunications, NanJing 210003, China

Abstract

Point cloud completion is a crucial task in the field of 3D imaging and is becoming a current research hotspot. However, traditional point cloud completion methods often fail to accurately restore the local details of objects due to the lack of detailed descriptions of the missing parts. This paper proposes a performance enhancement scheme for point cloud completion using tactile information. By employing a U-Net-based tactile feature extraction and fusion module, we enhance point cloud completion through the integration of tactile information. We validated our proposed network on the 3DVT dataset, and the experimental results demonstrate that our method effectively improves the performance of point cloud completion. Additionally, we conducted simulation experiments to mimic real-world scenarios where a robotic arm contacts objects, obtaining tactile point clouds using a mounted tactile simulator. These tactile point clouds were then used in related point cloud completion experiments, and the results were evaluated. The experimental results indicate that our proposed method significantly enhances the performance of point cloud completion and improves the local detail of the completed point clouds.

Keywords: Point cloud, Point cloud complete, Tacile assisted, Tacile fusion

1 Introduction

In order for robots to effectively interact with objects in their environment, it is essential that they possess an understanding of 3D space. Vision and touch are two important ways for robots to perceive their surroundings. Due to the widespread availability of low-cost lidar, depth cameras and single-photon 3D imaging, the use of 3D data in robotics is becoming increasingly common. Our team has previously proposed a novel multi-stage synergistic recovery network to reconstruct an accurate depth map in the field of single-photon 3D imaging [1]. However, in a cluttered environment, due to the inevitable occlusion between various objects, it is difficult to obtain the complete 3D data of objects in the scene. This can make it challenging to infer intermediate processes in downstream tasks such as robot grasping and path planning. On this issue, previous researchers have proposed to use RGB image or RGB-D image to reconstruct the shape of objects in 3D scenes [2, 3]. However, with the widespread use of cheap lidar and the progress of 3D point cloud algorithms, it is a better idea to use existing machine learning

algorithms to complete the incomplete point cloud obtained from a single view or in the case of occlusion. Because image loses a lot of 3D information [4], and the semantic attributes of image are usually implicit and entangled with each other, it is more difficult to reconstruct 3D shape with detailed semantic structures represented than reconstructing objects directly from 3D point clouds.

Thanks to the establishment of large-scale point cloud data sets, the point cloud completion methods based on deep learning have attracted more and more researchers' interest [5–10]. Machine learning-based methods often lose part of the input shape structure because they need to down-sample and encode the original point cloud. Therefore, they will use some methods to make up for the loss of details. For example, FBNet [11] uses a feedback-based point cloud completion network to reuse output information through feedback connections to enhance their own feature learning. SoftPool++ [12] proposes a novel convolution operator, which is used to learn the point cloud embedding in the encoder extracts permutation-invariant features from the point cloud via a soft-pooling of feature activations, and make it able to preserve fine-grained geometric details. PoinTr [13] is based on the set-to-set Transformer, and builds a local attention module in Transformer, which captures the geometric relationship in the point cloud by using the kNN model. But due to the compression in the encoder, a typical limitation of this type of architecture is that they tend to lose parts of the input shape structure. SnowflakeNet [14] stacks multiple Snowflake Point Deconvolution (SPD) modules to generate complete shapes, allowing missing point clouds to grow in 3D space like snowflakes. SeedFormer [15] introduces a new shape representation based on SnowflakeNet, called Patch Seeds, which captures the overall structure from partial inputs and preserves local region information. Although the above machine learning-based point cloud completion methods introduce many methods focusing on the local shape of the point cloud, these methods still do not well capture the geometric details and structures of local regions due to the lack of accurate local shape constraints. Here, we propose to utilize the structural information contained in the haptic point cloud to compensate the incomplete point cloud, and show its advantages. For example, in automated manufacturing and assembly processes, robots need to accurately perceive and handle the shapes and features of workpieces. By using tactile sensing to gather information about the surfaces of objects and combining it with incomplete point cloud data, more precise reconstruction of workpieces and assembly planning can be achieved, thereby improving production efficiency and product quality.

In recent years, the vision-based tactile sensor has been a hot research topic [16–19]. In the 3D shape reconstruction of the occluded objects, tactile sensor can be used to obtain the tactile information of the occluded part and make up the disadvantages of vision [20]. Because the vision-based tactile sensor can efficiently obtain tactile images, most of the existing studies on 3D reconstruction based on tactile information primarily rely on tactile images. Wang et al. [21] used a neural network with the ability to learn shape priors to first predict the 3D shape of an object from a single-view color image and then refine the shape with tactile perception. Smith et al. [20] used the GCN network to learn visual information from an RGB picture and project it onto the visual chart. At the same time, the tactile chart was predicted from the tactile information, and the iterative deformation of the chart was performed to obtain the global prediction. But all

of them acquire local information based on tactile images. Though 2D images contain implicit 3D information, it is much more difficult to gain the shape of 3D objects than point clouds, which inherently contain 3D spatial structure information.

This paper proposes a scheme for assisting point cloud completion using tactile point clouds. By simulating the interaction between a robotic arm and objects, tactile information is acquired through the DIGIT [22] sensor on the robotic arm, generating tactile point clouds. Then, using the U-Net-based tactile feature extraction and fusion scheme to perform the point cloud completion task by inputting both the tactile point clouds and the incomplete point clouds. Compared to 2D tactile images, tactile point clouds contain precise local geometric information about the object, addressing the limitations of conventional machine learning networks in capturing local details. This allows the network to better understand the scene comprehensively, enhancing the model's performance and generalization ability in point cloud completion tasks, and effectively improving the local details of the completed point clouds.

2 Approach

In this section, we introduce how to generate tactile point cloud and utilize the tactile point cloud to enhance the performance of point cloud completion. The overall framework is depicted in Fig. 1.

2.1 Tactile point cloud generation

In the simulation environment, we use a robotic arm with motorized grippers and each gripper has a DIGIT simulator for acquiring tactile images, as shown in Fig. 2. We simulate the process of obtaining the local shape information of objects. The local shape information comes from the depth information of the contact surface provided by the DIGIT sensor, which can impose strict constraints on the shape of the object and can help improve the visual prediction of the 3D shape.

During the simulation of touch, we put the target object at a fixed position in front of the robot arm, and the gripper was initially above the object. Then, we move the robot arm and control the closing of the jaws to contact the surface of the object until the DIGIT sensor generates touch image and depth image, as shown in Fig. 2. Then, we stop moving the robotic arm and record the depth information H , the position coordinate information x of the tactile sensor in space and rotation vector r .

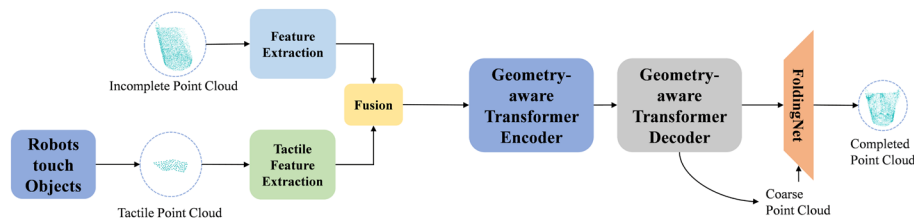


Fig. 1 Overall framework. The robot touches the object to acquire tactile information and generate tactile point clouds. These tactile point clouds are then fed into a network for tactile feature extraction. The extracted features are fused with incomplete point cloud features to enhance the point cloud completion process using tactile information

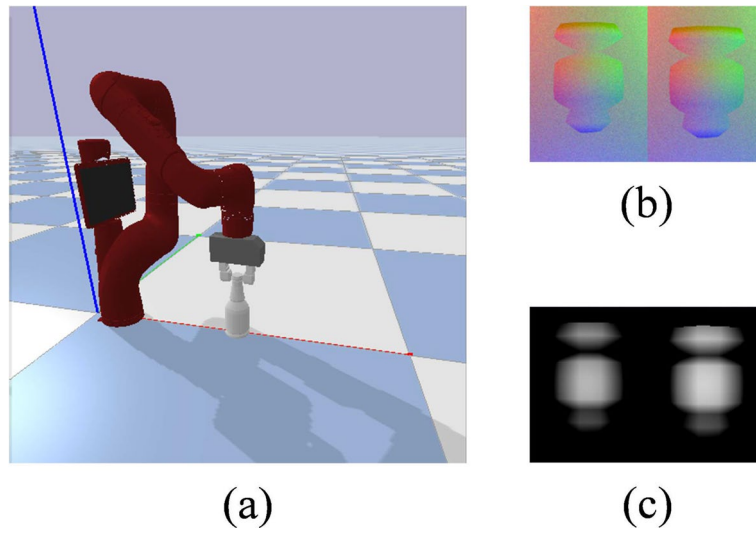


Fig. 2 **a** Object grasping; **b** Touch images of the tactile sensor; **c** Depth images

To generate the tactile point cloud, we initially establish a planar point cloud P_I at the origin of the robot's coordinate system and overlay the depth information obtained from the sensors onto this planar point cloud. Through this process, we can get a preliminary tactile point cloud P_c . Then we convert the obtained unit rotation vector r into an orthogonal rotation matrix R_t and get the final tactile point cloud P_t by rotating and translating the point cloud P_c :

$$\begin{aligned} P_t &= R_t * P_c + x, \\ P_c &= P_I + H, \\ R_t &= \cos\theta I + (1 - \cos\theta)rr^T + \sin\theta r^\wedge, \end{aligned} \quad (1)$$

where I is the identity matrix, r is the unit rotation vector, θ is the modulus length of the rotation vector, and $r^\wedge$ is the vector to the antisymmetric matrix operator, x is the tactile position.

2.2 Alignment of point cloud positions

We need to align the object coordinate system, including the robot coordinate system, and the world coordinate system to ensure that the object point cloud and the tactile point cloud are in the same coordinate system. We assume that the world coordinate system and robot coordinate system overlap, so we only need to align the object coordinate system and robot coordinate system. We can select the three points w_1 , w_2 and w_3 in the robot coordinate system and record the positions z_1 , z_2 , and z_3 of them in the object coordinate system, and then obtain the rotation matrix R_z and translation vector T_z by solving the linear equations:

$$\begin{aligned} X_w &= R_z * X_z + T_z, \\ X_z &= [z_1, z_2, z_3], \\ X_w &= [w_1, w_2, w_3]. \end{aligned} \quad (2)$$

In order to simplify the calculation process, we assume that the rotation matrix between the object and robot coordinate system is an identity matrix and we only need the translation vector between the two coordinate systems. Placing the object at a fixed position $w_0 = (0.5, 0, 0)^T$ in the x-axis direction of the robot coordinate system. For a point p_i on the object coordinate system, it easily corresponds to p_j in the robot coordinates system. We just need to add a translation to it:

$$p_j = p_i + w_0. \quad (3)$$

We can obtain the final tactile point cloud corresponding to the actual object by the conversion of the coordinate system, as shown in Fig. 3.

2.3 Visual and tactile feature extraction with integration and fusion

To better utilize the visual and tactile point clouds for the point cloud completion task, we have designed a feature extraction module based on U-Net [23] and DGCNN [24], as shown in Fig. 4. Tactile point clouds and incomplete point clouds undergo different feature extraction processes. Inspired by the application of U-net in the fields of point clouds and images, we employ cross layer connections to retain tactile information at different scales, utilizing this approach for initial feature extraction. Specifically, the feature extraction of touch points is down-sampled two times by the Max-Pooling layer and up-sampled two times by the deconvolution, and the skip connection is used at the same stage, which ensures that the final recovered feature map incorporates more low-level features and also enables the fusion of features of different scales point cloud features.

For incomplete point clouds, we use DGCNN network. DGCNN maintains the local geometry by building a local neighbor map, and then refines these local point cloud features by using EdgeConv, a process that enhances the representation of local and global features. Through the superposition of multiple EdgeConv, DGCNN gradually expands its receptive field, enabling it to capture complex point cloud structure and semantic information more comprehensively. Many approaches exploiting vision and touch for 3D tasks, especially in 3D reconstruction, tend to independently process visual and tactile information. Different from previous works that generate preliminary results of the final target and gradually refine them using tactile feedback, we prefer generating

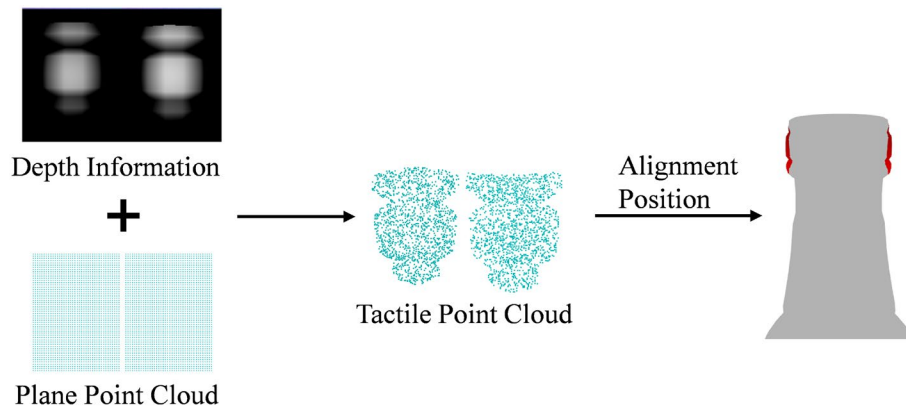


Fig. 3 The diagram for the tactile point cloud generation from the depth information and the plan point cloud after aligned in position

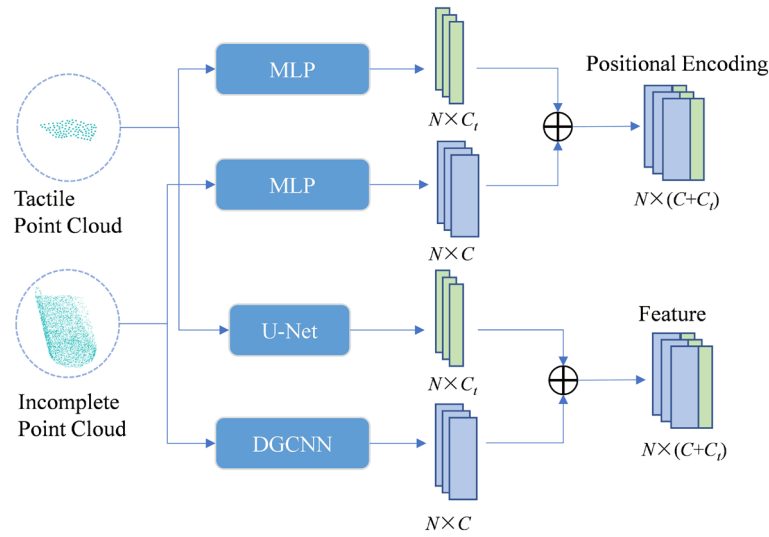


Fig. 4 U-Net and DGCNN feature extraction with integration and fusion

fused features to integrate both tactile and visual information. This approach enables us to comprehensively learn both the local and global features of tactile information, allowing touch information not only to enhance the upsampling at the touch site, but also to extrapolate its local neighborhood. Therefore, we connect the tactile point cloud features $F_t \in N \times C_t$, extracted through unet network with the incomplete point cloud features $F_p \in N \times C$, extracted through DGCNN network to obtain fusion features of $N \times (C_t + C)$.

In addition, we use multi-layer perceptron to encode the position of the tactile point cloud and the incomplete point cloud, and then concatenate to obtain final position encoding information of $N \times (C_t + C)$. By using fusion features of $N \times (C_t + C)$ and position encoding information $N \times (C_t + C)$, the model allow for a more complete understanding of point cloud data.

3 Results

In this section, we describe experiments that incorporate tactile point clouds during the point cloud completion process. We validated the effectiveness of the proposed U-Net-based tactile feature extraction and fusion scheme on the 3DVT (3D Visual and Touch) dataset and provided visual results of our method. Additionally, we simulated a robotic arm touching objects in a simulated environment to obtain tactile point clouds. Based on the acquired tactile point clouds, we conducted experiments and provided corresponding visual results. The complex models used in the simulated experiments were all selected from the ShapeNet [25] dataset.

The 3DVT dataset differs from commonly used point cloud datasets in current point cloud completion research, such as KITTI [26], ShapeNet, and PCN [27], as these datasets do not include tactile point clouds and therefore do not meet the requirements of this study. Smith [20] provided a comprehensive dataset that includes complete point clouds of object surfaces and multiple sets of tactile images and tactile point clouds based on the DIGIT tactile sensor. Building on this, we constructed a new dataset,

3DVT, to train the U-Net-based tactile feature extraction and fusion network proposed in this paper.

The 3DVT dataset contains a total of 12,732 objects. For each object in the dataset, there is a ground-truth point cloud with 8,192 points and 5 to 8 tactile point clouds from different touch positions. We split the 3DVT dataset into training and testing sets in a 4:1 ratio. Our proposed method was implemented in PyTorch and trained using the AdamW optimizer with an initial learning rate of 0.0005 and a weight decay of 0.0005. The training was conducted on an NVIDIA 3090 GPU.

3.1 Evaluation metric

According to some existing literature, we adopt the average chamfer distance as the evaluation criterion for the goodness of 3D point cloud reconstruction. Given the reconstructed point set P and the ground-truth point set G , the chamfer distance can be calculated as follows [27–30]:

$$d_{CD} = \frac{1}{|P|} \sum_{p \in P} \min_{g \in G} \|p - g\| + \frac{1}{|G|} \sum_{g \in G} \min_{p \in P} \|g - p\|. \quad (4)$$

L2 chamfer distance uses L2-norm to calculate the distance between two points.

3.2 Results on 3DVT

In our experiments, we conducted tests at three different scales: easy, medium, and difficult. Easy refers to an incomplete point cloud with a 25% missing ratio, medium refers to a 50% missing ratio, and difficult refers to a 75% missing ratio.

Table 1 shows the results of PoinTr not use tactile point cloud, directly concatenating the tactile point cloud with the incomplete point cloud, and using the U-Net-based tactile point cloud information extraction scheme. From the results in Table 1, it can be seen that directly concatenating the tactile point cloud with the incomplete point cloud and then feeding it into the network improves the performance compared to not using the tactile point cloud. However, the improvement is minimal. The U-Net-based tactile information extraction and fusion scheme significantly outperforms the direct concatenation of the tactile point cloud, achieving the best results when 25%, 50%, and 75% of the point cloud is missing.

Specifically, the completion results of the experiments are shown in Fig. 5. Fig. 5a is the real point cloud, Fig. 5b is the incomplete point cloud combined with the tactile point cloud, Fig. 5c is the completion result of the incomplete point cloud without the tactile point cloud. Fig. 5d shows the completion results using the U-Net-based tactile

Table 1 Experimental results on the 3DVT dataset

	CD-s	CD-m	CD-h
PoinTr [13]	0.825	1.258	2.537
Directly concatenate the tactile point cloud	0.818	1.234	2.446
The U-Net-based tactile information fusion and noise scheme	0.789	1.118	2.041
The U-Net-based tactile information fusion scheme	0.742	1.072	2.014

CD-s, CD-m, and CD-h are used to represent the chamfer distance results for easy, medium, and difficult conditions. The evaluation metric is L2 chamfer distance ($\times 1000$) (the lower the better (bold))

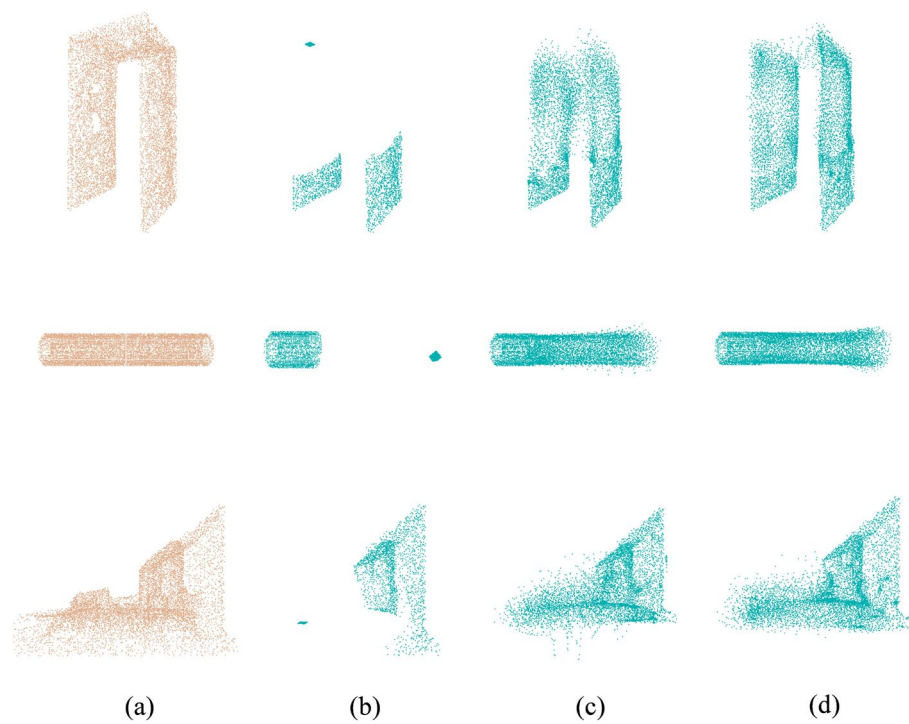


Fig. 5 Experimental results on the 3DVT dataset. **a** real point cloud; **b** incomplete point cloud with tactile point cloud; **c** completion results without using tactile information; **d** completion results using the U-Net-based tactile information fusion scheme

Table 2 Comparison between adding a tactile point cloud and non-tactile

Touch times	Sphere	Wine bottle	Cube	Bucket
0	1.8560	3.8789	0.5328	6.1621
1	1.1009	3.8155	0.5252	5.8977

The evaluation metric is L2 chamfer distance ($\times 1000$) (the lower the better (bold))

information fusion scheme. Compared to Fig. 5c, the U-Net-based tactile information extraction and fusion scheme successfully learns local detail information and restores the structure. The originally scattered local edge point cloud is reduced, the point cloud clusters together more closely, and the edges resemble the real point cloud more accurately.

3.3 Adding one tactile point cloud

To verify the effect of adding tactile point clouds on the point cloud completion task, we selected various types of objects from the ShapeNet dataset. We used simulated tactile sensors to obtain simulated tactile information in a simulated environment and generated tactile point clouds. We cropped the real point clouds of the objects into incomplete point clouds, then sent both the incomplete point clouds and the tactile point clouds into our network for point cloud completion. We compared the completion results before and after adding the tactile point clouds. The specific results are shown in Table 2, and the visualization results are shown in Fig. 6. As indicated in Table 2, adding tactile

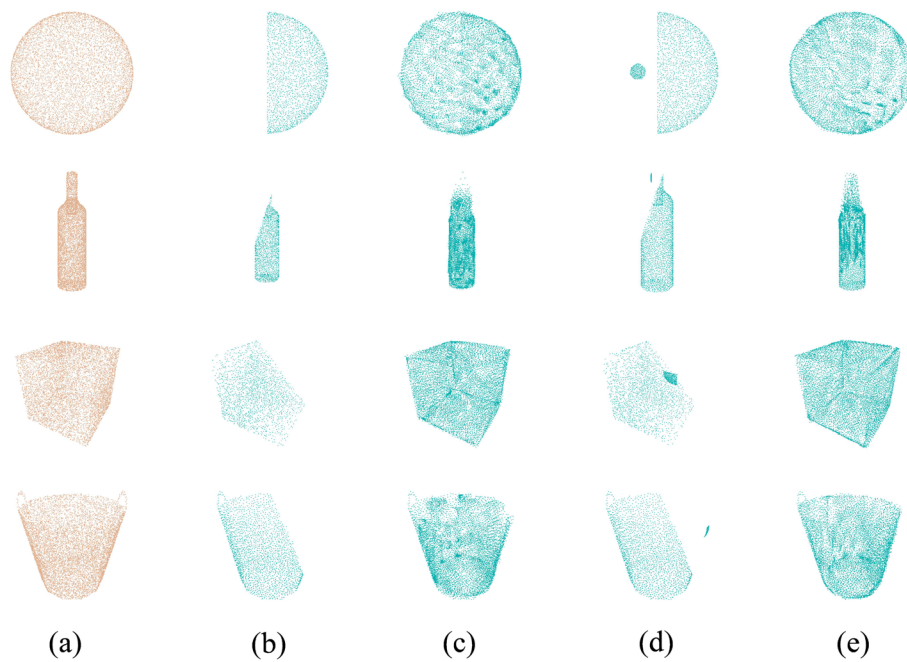


Fig. 6 Comparison between adding a tactile point cloud and non-tactile. **a** Ground Truth; **b** Incomplete point cloud; **c** Completion results corresponding to **(b)**; **d** Incomplete point cloud with a tactile point cloud; **e** Completion results corresponding to **(d)**

information improves the reconstruction results of the objects. For complex objects, such as buckets and wine bottles, adding tactile information is especially necessary because their shape details were not well reconstructed before adding the tactile point clouds. With tactile assistance, the completion results better match the actual shapes. Additionally, this also indirectly validates the accuracy of the simulated tactile point clouds. However, for some simple objects with very regular and symmetrical geometries (e.g., spheres, cubes), we can achieve very good completion results even without adding tactile point clouds, so the improvement brought by adding tactile information is not significant. Overall, considering that most practical applications involve reconstructing complex objects, adding tactile point clouds can effectively enhance the completion results in most scenarios.

3.4 Adding multiple touch

We selected various objects and added 1, 2, and 3 tactile point clouds to the incomplete point cloud, respectively. In Table 3, we discuss the impact of different numbers of tactile points on the completion results. It can be seen that as the number of tactile points increases, the reconstruction quality of most objects improves. In Fig. 7, we show the reconstruction results for wine bottles, guitars, and chairs. Without adding tactile information, the completed point clouds of these three models can depict the general shape but lack some details. For example, the wine bottle is missing details at the top, the chair is missing an armrest on the left side, and the point cloud on the right side of the guitar is somewhat scattered. When tactile point clouds are added to the wine bottle, the details at the top are better restored, and the originally scattered point clouds become

Table 3 Comparison of the effects by adding different times of tactile point clouds on the complete results

Category	0	1	2	3
Chair	3.4734	3.1455	2.6243	2.1366
Basket	2.7925	2.7378	2.4971	2.6884
Sphere	1.8560	1.1009	1.0739	1.0053
Wine bottle	3.8789	3.8155	3.8123	3.7968
Table lamp	8.1745	7.6285	7.5562	7.1588
Cube	0.5328	0.5252	0.5572	0.5452
Faucet	4.6039	4.1691	4.0752	3.9238
Bucket	6.1621	5.8977	5.7506	5.8957
Guitar	1.1697	1.1482	1.1020	0.9575

The evaluation metric is L2 chamfer distance ($\times 1000$) (the lower the better (bold))

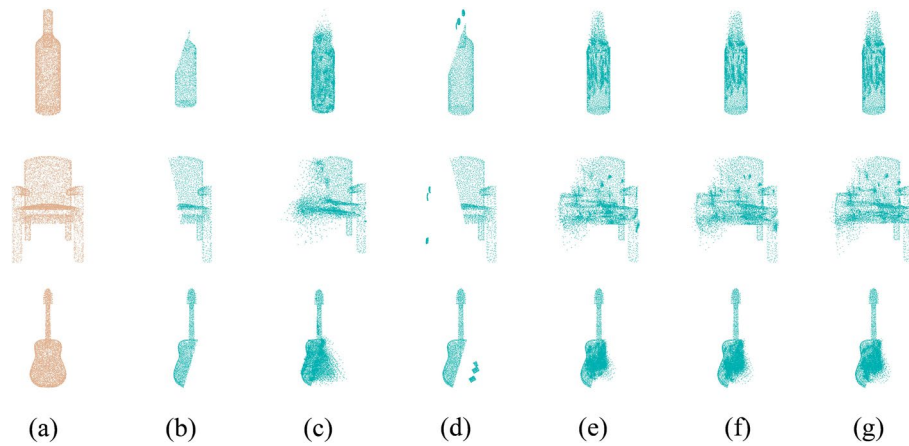


Fig. 7 Comparison of the effect by adding different numbers of touch point clouds on completion. **a** Ground Truth; **b** Incomplete point cloud; **c** Completion results corresponding to (d); **d** Incomplete point cloud with three tactile point clouds; **e** The completion result of an incomplete point cloud with a tactile point cloud; **f** The completion result of an incomplete point cloud with two tactile point clouds; **g** The completion result of an incomplete point cloud with three tactile point clouds

more concentrated as the number of tactile point clouds increases. After adding tactile point clouds to the armrest of the chair, the details of the armrest are better restored, and as more tactile point clouds are added, the edges also become clearer. For the guitar, the shape at the bottom significantly improves after adding tactile point clouds, with fewer scattered points at the edges, resulting in a clearer outline. In summary, as the number of tactile point clouds increases, the recovered local details more closely match the actual appearance. Additionally, when the original local point cloud is sparse and scattered, adding tactile point clouds can make the local point cloud denser and more convergent.

4 Conclusion

This study introduces a method to enhance point cloud completion with auxiliary tactile data. Local point clouds of objects are generated using digital tactile sensors in a simulated setting, followed by alignment with incomplete point clouds through calibration

between robotic and world coordinate systems. A feature extraction and fusion approach based on DGCNN and U-Net is then executed for point cloud completion, integrating both tactile and incomplete point clouds. To assess noise impact, a controlled amount of noise was added to the sensors. Subsequent experiments confirmed the method efficacy in improving completion quality, with potential applications in 3D imaging. In addition, we should declare that in this manuscript we have utilized simulation data for training and test. However, touching an object may cause some differences to simulation in practical applications. Therefore, in our future research work, we will take it into account and incorporate more realistic verification with our work.

Abbreviations

kNN	k-Nearest neighbor
SPD	Snowflake point deconvolution
fps	Farthest point sampling

Acknowledgements

The authors appreciate Mr. Z.A. Ren for enlightened discussions in haptic simulations. We gratefully acknowledge the financial support from Industrial Prospect and Key Core Technology Projects of Jiangsu provincial key R&D Program (BE2022071), National Natural Science Foundation of China (12074194, U19A2075), and the Leading-edge technology Program of Jiangsu Natural Science Foundation (BK20192001).

Author contributions

JHS: methodology, writing, original draft preparation, software. ML: methodology, writing, original draft preparation, software. JL: supervision, reviewing, methodology, validation. HRS: methodology, software. HYW: methodology, writing, software. CZ: supervision, methodology. PZW: supervision, methodology. QW: supervision, methodology, conceptualization. All authors read and approved the final manuscript.

Funding

Industrial Prospect and Key Core Technology Projects of Jiangsu provincial key R & D Program (BE2022071), National Natural Science Foundation of China (12074194, U19A2075), Leading-edge technology Program of Jiangsu Natural Science Foundation (BK20192001).

Availability of data and materials

The data will be available up on reasonable request.

Declarations

Competing interests

The Authors declare no conflict of interests.

Received: 13 January 2024 Accepted: 20 November 2024

Published online: 03 December 2024

References

1. Y.-H. Chen, J. Li, S.-P. Xie, Q. Wang, Single-photon 3d imaging with a multi-stage network. *Opt. Express* **30**(16), 29173–29188 (2022)
2. J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, J. Tenenbaum, M. Marnet: 3d shape reconstruction via 2.5 d sketches. *Adv. Neural Inf. Process. Syst.* **30** (2017)
3. D. Azinović, R. Martin-Brualla, D.B. Goldman, M. Nießner, J. Thies, Neural rgb-d surface reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6290–6301 (2022)
4. K. Fu, J. Peng, Q. He, H. Zhang, Single image 3d object reconstruction based on deep learning: a review. *Multimed. Tools Appl.* **80**(1), 463–498 (2021)
5. P. Mittal, Y.-C. Cheng, M. Singh, S. Tulsiani, Autosdf: shape priors for 3d completion, reconstruction and generation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 306–315 (2022)
6. G. Qian, H. Hammoud, G. Li, A. Thabet, B. Ghanem, Assanet: an anisotropic separable set abstraction for efficient point cloud representation learning. *Adv. Neural Inf. Process. Syst.* **34**, 28119–28130 (2021)
7. Y. Cai, K.-Y. Lin, C. Zhang, Q. Wang, X. Wang, H. Li, Learning a structured latent space for unsupervised point cloud completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5543–5553 (2022)
8. D. Maturana, S. Scherer, Voxnet: a 3d convolutional neural network for real-time object recognition. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015) pp. 922–928.

9. C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L.J. Guibas, Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5648–5656 (2016)
10. C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
11. X. Yan, H. Yan, J. Wang, H. Du, Z. Wu, D. Xie, S. Pu, L. Lu, Fbnet: Feedback network for point cloud completion. In: European Conference on Computer Vision, pp. 676–693 (2022). Springer
12. Y. Wang, D.J. Tan, N. Navab, F. Tombari, Softpool++: an encoder-decoder network for point cloud completion. *Int. J. Comput. Vis.* **130**(5), 1145–1164 (2022)
13. X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, J. Zhou, Pointn: diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12498–12507 (2021)
14. P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, Z. Han, Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5499–5509 (2021)
15. H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, C. Wang, Seedformer: Patch seeds based point cloud completion with upsample transformer. In: European Conference on Computer Vision, (Springer, 2022) pp. 416–432.
16. Z. Kappassov, J.-A. Corrales, V. Perdureau, Tactile sensing in dexterous robot hands. *Robot. Auton. Syst.* **74**, 195–220 (2015)
17. M.K. Johnson, F. Cole, A. Raj, E.H. Adelson, Microgeometry capture using an elastomeric sensor. *ACM Trans. Graph. (TOG)* **30**(4), 1–8 (2011)
18. S. Dong, W. Yuan, E.H. Adelson, Improved gelsight tactile sensor for measuring geometry and slip. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (IEEE, 2017) pp. 137–144.
19. R. Li, R. Platt, W. Yuan, A. Ten Pas, N. Roscup, M.A. Srinivasan, E. Adelson, Localization and manipulation of small parts using gelsight tactile sensing. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IEEE, 2014) pp. 3988–3993 (2014).
20. E. Smith, R. Calandra, A. Romero, G. Gkioxari, D. Meger, J. Malik, M. Drozdal, 3d shape reconstruction from vision and touch. *Adv. Neural Inf. Process. Syst.* **33**, 14193–14206 (2020)
21. S. Wang, J. Wu, X. Sun, W. Yuan, W.T. Freeman, J.B. Tenenbaum, E.H. Adelson, 3d shape perception from monocular vision, touch, and shape priors. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (IEEE, 2018) pp. 1606–1613.
22. M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V.R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, Digit: a novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robot. Autom. Lett.* **5**(3), 3838–3845 (2020)
23. O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-assisted intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, (Springer, 2015) pp. 234–241.
24. A.V. Phan, M. Le Nguyen, Y.L.H. Nguyen, L.T. Bui, Dgcnn: a convolutional neural network over large-scale labeled graphs. *Neural Netw.* **108**, 533–543 (2018)
25. Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
26. A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset. *Int. J. Robot. Res.* **32**(11), 1231–1237 (2013)
27. W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV), (IEEE, 2018) pp. 728–737.
28. L.P. Tchammi, V. Kosaraju, H. Rezatofighi, I. Reid, S. Savarese, Topnet: structural point cloud decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 383–392 (2019)
29. H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, W. Sun, Grnet: Gridding residual network for dense point cloud completion. In: European Conference on Computer Vision, (Springer, 2020) pp. 365–381.
30. B. Fei, W. Yang, W.-M. Chen, Z. Li, Y. Li, T. Ma, X. Hu, L. Ma, Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. *IEEE Trans. Intell. Transport. Syst.* **23**(12), 22862–22883 (2022)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.