A REMOTE SHARING METHOD OF 3D PHYSICAL OBJECTS WITH INSTANCE-SEGMENTED 3D POINT CLOUD ACQUIRED IN REAL TIME

RYO ONISHI[1], TOMOHIRO FUKUDA[2] and NOBUYOSHI YABUKI[3]
[1,2,3]*Osaka University.*
[1]*onishi@it.see.eng.osaka-u.ac.jp, 0000-0003-10130-7308*
[2]*fukuda.tomohiro.see.eng@osaka-u.ac.jp, 0000-0002-4271-4445*
[3]*yabuki@see.eng.osaka-u.ac.jp, 0000-0002-2944-4540*

Abstract. In the field of architecture and urban design, physical models are used in design meetings. Furthermore, teleconferencing via the internet has begun to be widely used in society due to COVID-19 and in preparation for disasters. Although conventional web conferencing can share only 2D information through screens, it is expected that interactive screen sharing of physical objects will enable smoother remote conferencing. A system that can manipulate point clouds in clusters by dividing real-time point clouds captured from 3D real objects by distance has been reported as a way to share physical objects. However, because the point clouds are divided by distance between the two clusters when the point clouds get closer than some threshold, they become treated as a single object. In this study, we aim to develop a system that uses instance segmentation to divide point clouds by region rather than by distance between objects. This system is expected to contribute to the realisation of better architectural and urban design processes without any misunderstandings among the parties involved and to the reduction of unnecessary energy consumption due to travel for face-to-face meetings.

Keywords. Remote Meeting; Fast Point Cloud; Instance Segmentation; Three-dimensional Remote Sharing; Mixed Reality; SDG 11; SDG 13.

1. Introduction

In the fields of architecture and urban design, physical models such as design study models and building material samples are used in design meetings to help stakeholders, including experts and non-experts, understand the content of a project. Furthermore, remote meetings via the internet and hybrid meetings that combine these two methods are beginning to be widely used in society in addition to traditional face-to-face meetings as a result of COVID-19 and in preparation for disasters. Although conventional web conferencing can share only 2D information such as images and video via a screen, there is a demand for interactively sharing physical objects on the screen to enable smoother remote conferencing.

POST-CARBON, Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2022, Volume 2, 395-404.

3D virtual models can be created using computer-aided design and building information modelling and shared over a network from remote locations. However, creating a 3D virtual model from scratch is time-consuming and sharing a physical object that moves and deforms is difficult. A point cloud is a set of points with coordinates and colour information that can be acquired using a 3D scanner or RGB-D camera. As a system for sharing point clouds, a client-server system has been developed that captures a static 3D physical scene in real time and allows a large number of users to explore it (Stotko et al., 2019).

However, although this system can build 3D virtual models in mixed reality (MR) scene, it cannot manipulate them. Therefore, a system that can manipulate point clouds in clusters by using fast point cloud segmentation has been reported (Ishikawa et al., 2020). However, the Euclidean cluster method used in this study divides the point cloud by the distance between two clusters, so if the distance between point clouds is below a certain threshold, they are treated as the same.

The purpose of this research was to develop a system that uses instance segmentation to divide a real-time point cloud captured from a real 3D object by the area of the object rather than the distance between objects. This system displays the sender object and its physical object in MR as a segmented point cloud, even if the sender touches a shared physical object, and the receiver can manipulate the point cloud remotely and individually. We experimentally analysed the relationship between the number of points in the transmitted point clouds and the frames per second (fps) at which the point clouds are displayed in MR, and examined the number of point clouds that can be transmitted in real time, as well as down-sampling point clouds to make it possible to remotely share more objects in real time. This system allows the receiver to freely move and rearrange the shared objects on the sender's side, thus facilitating a teleconference for design review while communicating with the sender. This system will enable a better architectural and urban design process while resolving misunderstandings between the parties involved, contributing to SDG 11. In addition, Greenhouse Gas (GHG) emissions generated by the transportation sector are increasing rapidly, especially in emerging countries (Yan and Crookes, 2009). Smooth video conferencing can contribute to SDG 13 by facilitating remote work and reducing GHG emissions associated with commuting.

2. Previous Research

2.1. TELEPRESENCE

Telepresence (Minsky, 1980) is a technology that allows people to experience a sense of presence as if they were sharing the same space face-to-face while in a remote location. Currently, many immersive telepresence systems are being developed using augmented reality and MR, such as head-mounted displays (HMDs), to share virtual 3D information. One such system has been developed that can capture a person in a space surrounded by cameras and display an elaborate 3D avatar in a virtual space (Petit et al., 2010). However, this system is difficult to use for teleconferencing because it requires large-scale equipment. Later, the widespread availability of inexpensive RGB-D cameras such as Microsoft Kinect has led to the development of room-scale virtual space sharing. Client-server systems have been developed to capture static 3D

scenes in real time and allow a large number of users to explore them (Stotko et al., 2019). However, although such systems can build 3D models in MR, they cannot manipulate them. A system that can manipulate the created point clouds by using fast point cloud segmentation has also been developed (Ishikawa et al., 2020). However, the Euclidean cluster method used in this research divides a point cloud based on the distance between two points, so if the distance is less than a certain threshold, the points are processed as the same point cloud.

2.2. INSTANCE SEGMENTATION

Instance segmentation can classify objects of the same class and solve the problem of overlapping objects of the same class by detecting object candidate regions and creating mask images for each candidate region. It also solves the problem of overlapping classes. Because of its importance, a lot of research has been performed to improve the accuracy of instance segmentation. Mask R-CNN (He et al., 2017) is a typical two-stage instance segmentation method in which candidate regions of interest are first generated and then classified and segmented. However, this two-stage method requires reporting of features for each region of interest and processing in subsequent computations, which does not allow for real-time performance even when the image size is small. Although fast processing semantic segmentation also exists, there is still little research that focuses on fast processing instance segmentation. However, systems able to perform fast instance segmentation by performing detection and identification in parallel have been proposed, including YOLACT (Bolya et al., 2019a) and YOLACT++ (Bolya et al., 2019b).

2.3. POINT CLOUD SEGMENTATION

The conditions for inputting point cloud data into a convolutional neural network are that changing the order of the point cloud does not change the result, the relationship between close points must be preserved, and rotation and translation do not change the result. However, PointNet (Qi et al., 2017) was developed to improve the problem by segmenting the point cloud as-is. In addition, PointNet++ (Qi et al., 2019), which can recognise point clouds regardless of their density scales, was also developed. However, this method has problems, such as not being able to handle large-scale point clouds. Highly accurate point cloud segmentation methods such as 3D-BoNet (Yang et al., 2019) have also been reported, but these do not offer sufficient processing speed for real-time capability. For fast 3D segmentation, there is RangeNet++ (Milioto et al., 2019), SalsaNext (Cortinhal et al., 2020) and others. However, these are semantic segmentation, not instance segmentation, and cannot be used in this research.

In this study, we use YOLACT++, which is a fast image instance segmentation method that can operate on multiple real objects of the same class, to segment the RGB image before creating the point cloud and then classify the point cloud after combining it with the depth image by category for point cloud segmentation.

3. Proposed Method

An overview of the system is shown in Figure 1. We propose a remote-sharing system that allows a receiver wearing an MR-HMD to manipulate the point cloud of a 3D

object from real-time point clouds captured at a remote location. This method can be used at two separate locations (Sites A and B) connected by a local area network (LAN). At Site A, an RGB-D camera continuously generates a real-time point cloud of the shared real object, classifies the points into individual clusters, and stores the location, colour, and cluster labels of the point cloud data. At Site B, the MR-HMD acquires and renders the transferred point cloud data, allowing the recipient to manipulate objects individually.

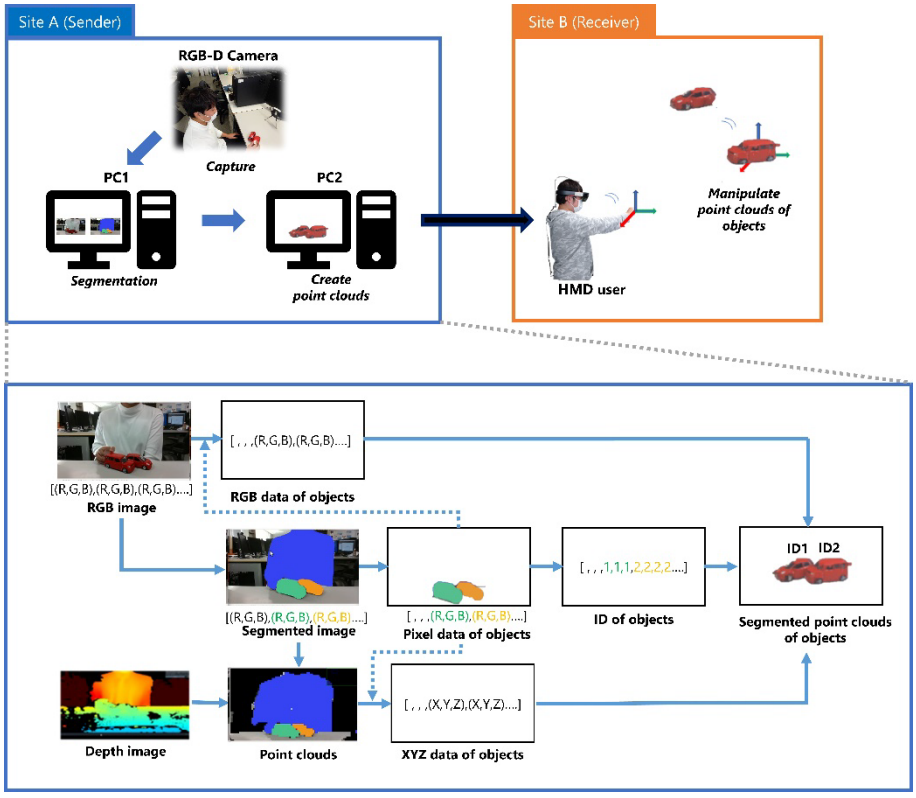

Figure 1. System overview

3.1. POINT CLOUD SEGMENTATION PROCESS

Point cloud processing first performs instance segmentation on the RGB images acquired by the RGB-D camera to obtain an array of 2D images of the object regions of the shared real object or person. Next, the RGB values of the object regions are obtained from the array of RGB images against the array information of each object region obtained, and the xyz values of the object regions are obtained from the array of point cloud images. Cluster labels are then assigned to each object region, and the data are sent to PC2 to be combined to create a point cloud segmented by instance. The real-time point cloud is then transferred to the MR-HMD at Site B via wireless a LAN as point cloud data which includes coordinates, colours, and cluster labels. The number

of points in the point clouds to be created afterwards can be reduced by thinning out the 2D image array of the object region extracted by segmentation, thereby decreasing the rendering fps of the point cloud.

3.2. PROCESS OF DISPLAY AND OPERATION IN MR

For MR display and manipulation, the MR-HMD receives the point cloud data and draws the virtual object as a point cloud in the real world. The point cloud has cluster labels so that each virtual object can be identified individually, and the HMD user can use hand gestures to move each segmented point cloud up, down, left, right, forward, and backward, individually, using a pinch motion that requires the thumb and forefinger of one hand to close. In order for the viewpoint cursor to appear on the object, collision detection is used, which requires a mesh for the object. However, since it is difficult to accurately mesh the surface of a 3D point cloud, we use a convex hull to create an invisible mesh that allows pinch operations to be used.

4. Verification

In this section, we describe the verification experiments conducted to evaluate the proposed method. Section 4.2 verifies whether the proposed method can segment and manipulate objects even if they are close together. In section 4.3, we analyse the relationship between the number of point clouds to be sent and the fps at which the point clouds are displayed in MR, and verify the number of point clouds that can be sent in real time.

4.1. BUILDING A PROTOTYPE SYSTEM

We used an Intel RealSense D435 as the RGB-D camera. On PC1 running Ubuntu OS, an Intel RealSense D435i was used for RGB image, depth image acquisition, point cloud creation using the Intel RealSense software development kit (SDK), and segmentation using YOLACT ++. On PC2 running Windows OS, Visual Studio was used for MR rendering, Unity was used for output to MR-HMD, and HoloToolkit was used for frame rate measurement of the MR rendering process and point cloud manipulation. Microsoft HoloLens was used as the HMD for experiencing MR.

4.2. VERIFICATION OF OPERATION OF THE PROPOSED METHOD

The system was tested in two rooms connected by a wireless LAN. Figure 2 shows the positions of the sender, who captures the shared object using the RGB-D camera, and the receiver, who wears the HMD. In order to check whether segmentation is possible even when the sender touches the shared object, segmentation is performed while the sender touches a toy car. The results of the sender and receiver are shown in Figure 3. On the sender's side, the sender and the two vehicles are split exactly. In MR, the red car toy is extracted from the sender's hand, and only the two shared objects are transmitted. When the sender pinches the red car in the MR, only the red car in MR approaches the sender. This suggests that the point cloud of objects of the same class

at a short distance can also be divided and manipulated.

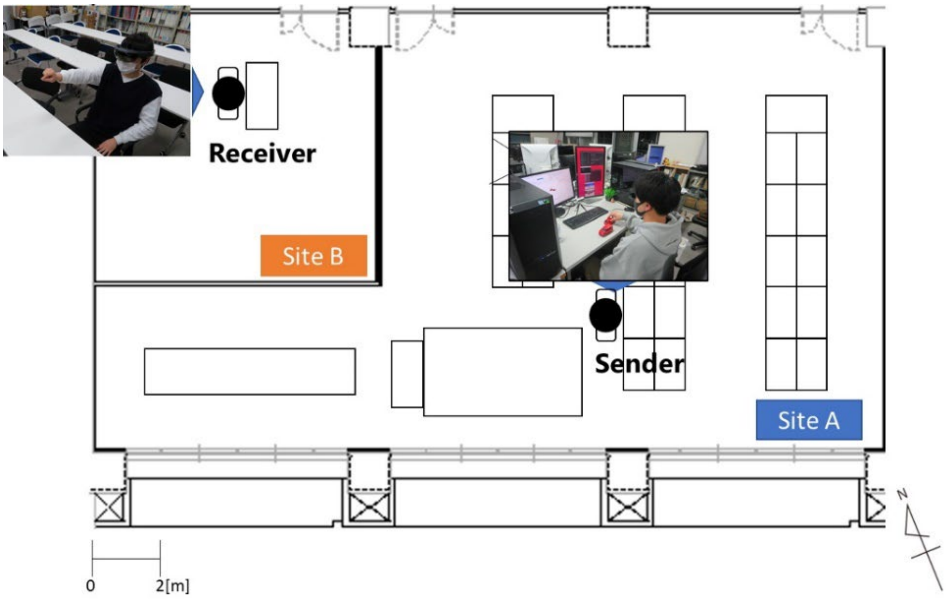

Figure 2. Verification environment

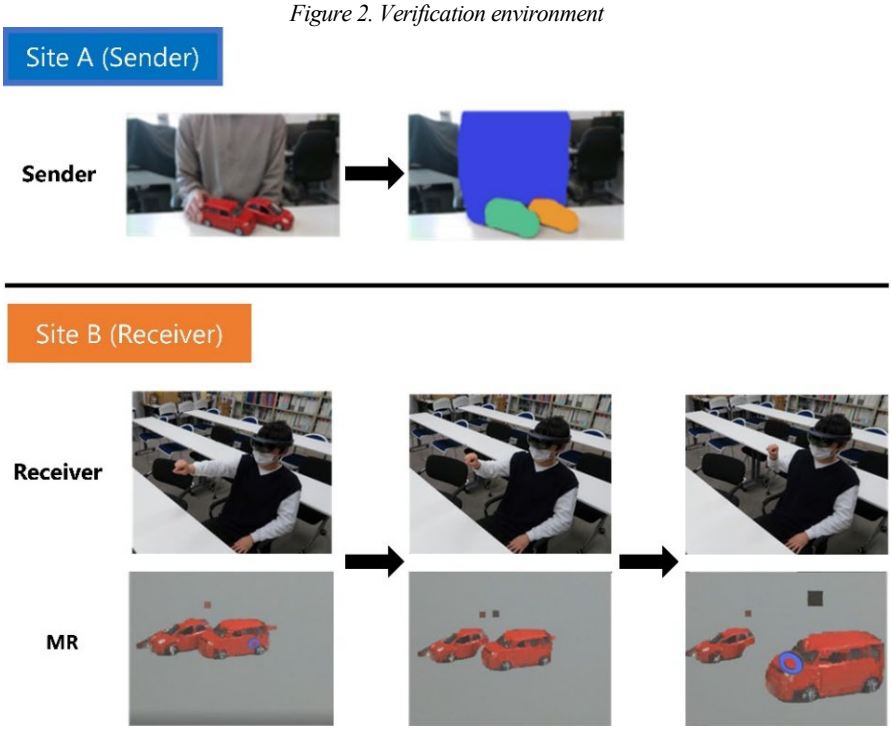

Figure 3. Verification result

4.3. EVALUATION OF POINT CLOUD DATA COMMUNICATION

In order to verify the relationship between the number of points in the point clouds and the fps rendered on the HMD, we increased the number of transmitted point clouds by 1000 units and examined the fps of rendering the point clouds in MR corresponding to each number of points 50 times. The relationship between these point clouds and the fps is shown in Figure 4. As shown in the figure, the coefficient of determination R exceeds 0.7 when expressed using the power approximation, so these values may be used for verification. From this figure, we confirmed that the number of points in the point clouds at 15 fps, which ensures real-time performance, is less than 4000 points, and after 5000 points, the rendering rate decreases to less than 15 fps, which cannot be said to have real-time performance (Chen and Thropp, 2007).

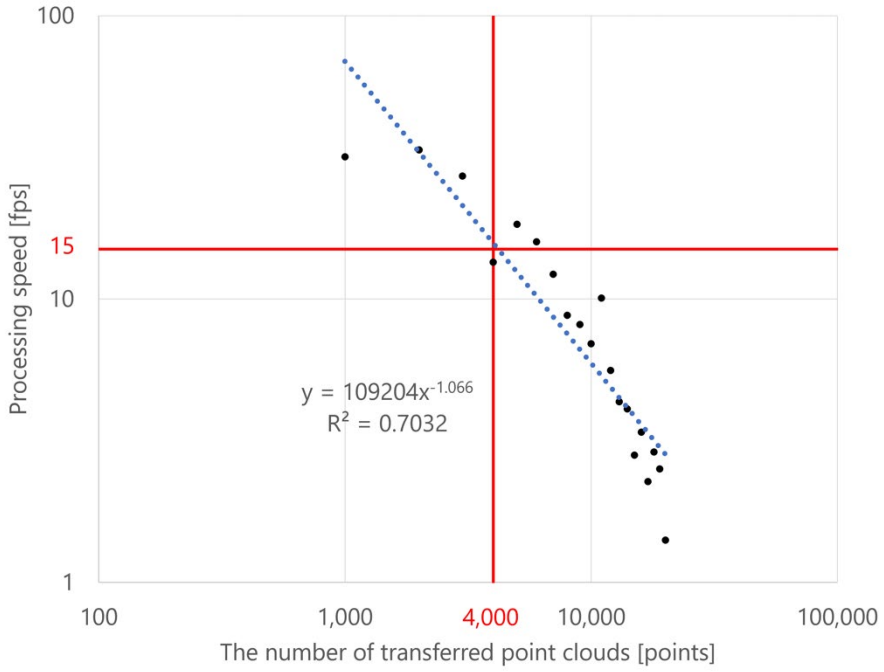

Figure 4. Relationship between rendering fps and number of points in point clouds

Next, the number of points in point clouds with about 12,000 points for the two shared objects was reduced to 1/2, 1/3, 1/4, and 1/5 by down-sampling the point clouds by thinning the 2D array. The measured average of the number of points in the point clouds and rendering fps is shown in Figures 5 and 6. The measurement results show that when the number of points in point clouds was reduced to 1/4 and 1/5, the FPS exceeded 15 fps and real-time performance was achieved. In addition, since the down-sampling process reduced the rendering fps, the difference in the number of points in point clouds that can ensure real-time performance was confirmed.

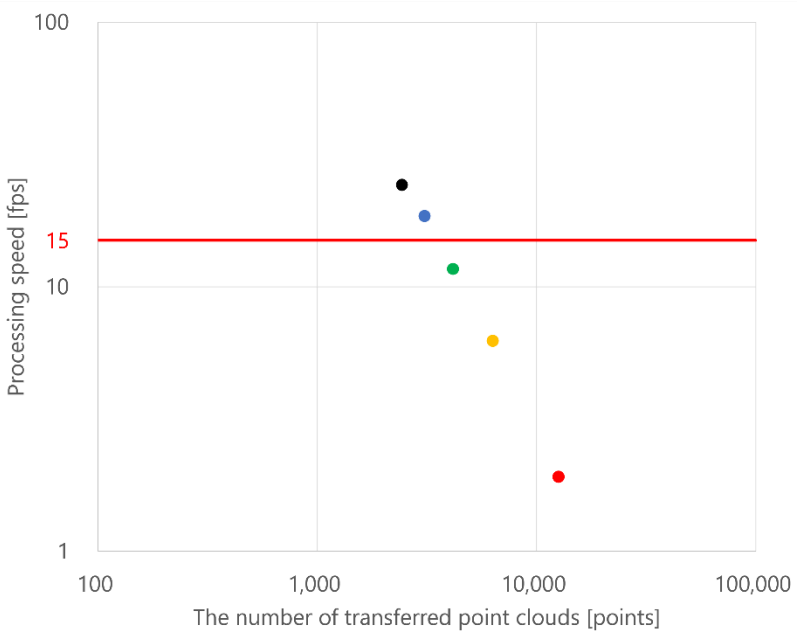

Figure 5. Relationship between rendering fps and number of points in point clouds (down sampling)

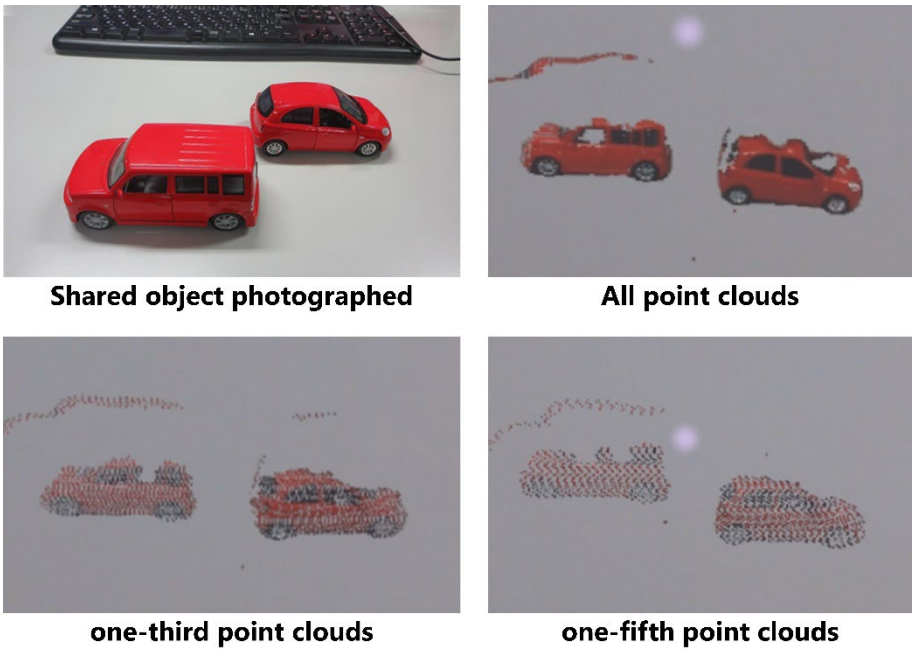

Figure 6. Down sampling result

5. Discussion

Verification of the operation of the proposed method found that the sender can extract only shared objects by dividing 3D real objects into object regions in real time by instance segmentation. On the receiver side, it was confirmed that the real-time point cloud could be segmented and manipulated individually even for objects of the same class separated by a short distance. By measuring the relationship between the number of points in the point clouds and the drawing fps, we found that the number of points in the point clouds that can ensure real-time performance is less than 4,000, and multiple objects are difficult to share. To solve this problem, we down-sampled the point clouds created by thinning out the 2D image array, and were able to render and manipulate multiple objects in MR in real time. As a result, we were able to realise a system for sharing and manipulating 3D real objects in a remote location in real time, regardless of the distance between the objects.

One issue is that if the number of points in point clouds is greatly reduced, the visibility of the point clouds of the shared object is reduced. This issue is expected to be improved in the future when the amount of data that can be transmitted is greatly increased as emerging technologies such as 5G become more widespread. Another issue was that point clouds of overlapping objects and areas not visible to the camera were missing. This problem can be solved by using multiple cameras to capture images from different angles and then combining the point clouds to create a complete point cloud of the 3D real object without any missing points.

6. Conclusion

In this study, we proposed a system for remotely sharing 3D real objects such as models that can be easily understood by stakeholders in architecture and urban design projects. Previous studies (Ishikawa et al., 2020) have raised the issue that segmentation cannot be performed when objects are closer than some threshold value. To solve this problem, we designed a system that creates a point cloud segmented by each object using instance segmentation of the point cloud and allows the object to be manipulated individually in MR even if it is touched or moved by a person, independent of distance. Objects of the same class can also be classified by instance segmentation and can be manipulated individually in MR. Even when multiple objects are shared, instance segmentation removes non-object point clouds, and down-sampling further reduces the number of point clouds, enabling real-time rendering. Issues related to when the sender's shared object is hidden and when the point cloud in the area that cannot be captured by the RGB-D camera is missing will be addressed in future work.

Acknowledgements

This research has been partly supported by the research grant of Nohmura Foundation for Membrane Structure’s Technology.

References

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019a). YOLACT: Real-time instance segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision8ICCV* (pp. 9157-9166).

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019b). YOLACT++: Better Real-time Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2), 1108-1121. https://doi.org/ 10.1109/TPAMI.2020.3014297.

Chen, J. Y. C., & Thropp, J. E. (2007). Review of Low Frame Rate Effects on Human Performance; *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, 37(6), 1063-1076.

Cortinhal, T., Tzelepis, G., & Aksoy, E. E. (2020). SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds, *Lecture Notes in Computer Science*, 12510, 207-222.

He, K., Gkioxari, G., & Dollar, P. (2017). Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV) 2017* (pp. 2980–2988). https://doi.org/10.1109/ICCV.2017.322.

Ishikawa, D., Fukuda, T., & Yabuki, N. (2020). A Mixed Reality Coordinate System for Multiple HMD Users Manipulating Real-time Point Cloud Objects - Towards virtual and interactive 3D synchronous sharing of physical objects in teleconference during design study. *Proceedings of the 38th eCAADe Conference* (pp. 197-206).

Live Telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 25(5), 2102-2112. https://doi.org/10.1109/TVCG.2019.2899231.

Milioto, A., Vizzo, I., Behley, J., & Stachniss, C. (2019) RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. *IEEE International Conference on Intelligent Robots and Systems* (pp. 4213-4220). https://doi.org/10.1109/IROS40897.2019.8967762.

Minsky, M. (1980, June). Telepresence*. OMNI Magazine*, 44-52.

Petit, B., Lesage, J.-D., Menier, C., Allard, J., Franco, J.-S., Raffin, B., Boyer, E. & Faure, F. (2010), Multicamera Real-Time 3D Modeling for Telepresence and Remote Collaboration. International, *Journal of Digital Multimedia Broadcasting* 2010, 247108. https://doi.org/10.1155/2010/247108

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 652-660).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. *In 31st Advances in Neural Information Processing Systems (NIPS 2017)* (pp. 5099-5108).

Stotko, P., Krumpen, S., Hullin, M. B., Weinmann, M., & Klein, R. (2019). SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence. *IEEE transactions on visualization and computer graphics*, 25(5), 2102-2112.

Yan, X., & Crookes, R. J. (2009). Reduction potentials of energy demand and GHG emissions in China's road transport sector. *Energy Policy*, 37(2), 658-668. https://doi.org/10.1016/j.enpol.2008.10.008

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., & Trigoni, N. (20). Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In *33rd International Conference on Neural Information Processing Systems* (pp. 2940-2949).