Generating 3D Digital Twins of Real Indoor Spaces based on Real-World Point Cloud Data

Wonseop Shin[1], Jaeseok Yoo[2], Bumsoo Kim[3], Yonghoon Jung[3], Muhammad Sajjad[4], Youngsup Park[5], and Sanghyun Seo[3,6*]

[1] Graduated School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University, Seoul, South Korea
[e-mail: wonseop218@cau.ac.kr]
[2] Nextchip, Seoul, South Korea
[e-mail: jaeseokyoo9599@gmail.com]
[3] Department of Applied Art and Technology, Chung-Ang University, Anseong, South Korea
[e-mail: bumsookim@cau.ac.kr, dydgns2017@cau.ac.kr]
[4] Digital Image Processing Laboratory, Department of Computer Science, Islamia College University Peshawar, Peshawar 25000, Pakistan
[e-mail: muhammad.sajjad@icp.edu.pk]
[5] INNOSIMULATION CO., LTD, Seoul, South Korea
[e-mail: yspark@innosim.com]
[6] College of Art and Technology, Chung-Ang University, Anseong, South Korea
[e-mail: sanghyun@cau.ac.kr]
[*]Corresponding author: Sanghyun Seo

Received March 28, 2024; accepted June 3, 2024; published August 31, 2024

Abstracts

The construction of virtual indoor spaces is crucial for the development of metaverses, virtual production, and other 3D content domains. Traditional methods for creating these spaces are often cost-prohibitive and labor-intensive. To address these challenges, we present a pipeline for generating digital twins of real indoor environments from RGB-D camera-scanned data. Our pipeline synergizes space structure estimation, 3D object detection, and the inpainting of missing areas, utilizing deep learning technologies to automate the creation process. Specifically, we apply deep learning models for object recognition and area inpainting, significantly enhancing the accuracy and efficiency of virtual space construction. Our approach minimizes manual labor and reduces costs, paving the way for the creation of metaverse spaces that closely mimic real-world environments. Experimental results demonstrate the effectiveness of our deep learning applications in overcoming traditional obstacles in digital twin creation, offering high-fidelity digital replicas of indoor spaces. This advancement opens for immersive and realistic virtual content creation, showcasing the potential of deep learning in the field of virtual space construction.

Keywords: Digital twin, Deep learning, 3D reconstruction, Image inpainting, 3D object detection, Virtual space construction.

A preliminary version of this paper was presented at ICONI 2023, and was selected as an outstanding paper. This research work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2022R1A2C1004657).

1. Introduction

The surge in interest surrounding virtual content technology stems from the swift advancement of remote activities, Virtual Reality (VR) / Augmented Reality (AR) hardware devices, and the maturation of 5G network technology. In 3D virtual spaces, economic, social, and cultural activities take place through interactions between people, leading to the growing utility of virtual spaces like the metaverse. To make users interested in and easily immersed in metaverse spaces, it is necessary to construct virtual spaces that accurately reflect real-world information, such as spatial layouts and objects.

The two main methods for building virtual spaces are 3D modeling using authoring tools and video-based approaches using cameras. By utilizing 3D authoring software tools like Maya, 3D Max, and SketchUp, designers can directly model real-world information, encompassing objects, spaces, and more. This method has the advantage of creating noise-free virtual spaces. However, the resultant quality may vary depending on the proficiency of the designer, and it may entail significant costs and time investment [1]. Recent studies have focused on creating three-dimensional reconstructions of real-world spaces based on videos captured by cameras [2], which involves calculating the transformation relationships (feature points) between different viewpoints and extracting point clouds using depth information. This allows for the automatic reconstruction of realistic virtual spaces containing real-world spatial data (lighting, materials, etc.).

However, point clouds frequently exhibit a notable amount of noise, and there is a potential for information loss in areas that remained unrecorded or were obscured by objects. This is particularly true for indoor spaces, which are complex and contain various objects that obscure wall surfaces. As a result, when extracting indoor point clouds, some areas are lost, and objects composed of point clouds are susceptible to increased noise levels as the complexity of the indoor space increases. Indoor environments particularly include reflective materials like glass and aluminum, and objects within the space also contain such information. This poses challenges during the extraction of point clouds from videos captured by RGB-D cameras, resulting in significant noise in the reconstructed space and objects. Moreover, the diverse and complex objects present in indoor spaces obscure information about wall surfaces and floor planes. Even with precise capture facilitated by RGB-D cameras, the extraction of point clouds for occluded areas remains unattainable, and there will persist unobserved areas, particularly within complex and spacious indoor environments. Consequently, the problem of point cloud vanishing arises.

Concerning this, active research on automating the classification of 3D objects in indoor point clouds utilizes clustering and deep learning techniques [3-6], while other studies utilize interpolation and GAN-based methods to restore lost areas in point clouds [7-9]. However, in present studies, these tasks often occur as distinct processes, making it challenging to construct virtual spaces that closely resemble the real world. In response, this paper proposes a pipeline for constructing indoor virtual spaces by estimating the space structure from images obtained through a widely available device, the RGB-D camera, and utilizing 3D object recognition and lost area inpainting techniques. The proposed method minimizes manual work and enables the efficient creation of virtual indoor spaces. Unlike traditional point cloud-based indoor reconstruction methods, it can generate a 3D representation of occluded areas. Additionally, it overcomes the challenges encountered in automating the construction of indoor spaces and produces spaces that are easily applicable to future virtual content. The contributions of this paper are as follows:

■ The proposed pipeline can efficiently construct virtual indoor spaces, minimizing manual intervention.

■ The method can generate a 3D representation of occluded areas, overcoming limitations of existing point cloud-based indoor reconstruction methods.

■ The created spaces are suitable for utilization in future virtual content, providing practical benefits.

2. Related Work

We aim to extract point clouds from images captured by an RGB-D camera and use them to reconstruct indoor spaces. Recent research has been conducted on 3D reconstruction techniques and indoor space structure estimation based on indoor point clouds. In this section, we explore existing methods that are relevant to our research.

2.1 3D Reconstruction

3D reconstruction techniques are essential for automating the construction of virtual spaces that closely resemble the real world. Research in this area primarily focuses on extracting point clouds from images. However, challenges such as indoor space characteristics (reflective materials, complexity), occlusion caused by objects, and unobserved areas lead to noise and information loss in virtual spaces. To address these challenges, research is being conducted on modeling based on the space structure. One method for generating point clouds from images is the stereo vision approach, which is based on the principles of human binocular vision. Stereo vision enables the simultaneous acquisition of depth and color information and primarily utilizes affordable RGB-D cameras rather than laser scanning devices. Point clouds can be extracted by leveraging depth images that contain distance information between colors and objects. Since the publication of the study by Richard A. et al. [10], there has been active research on using RGB-D cameras to reconstruct indoor spaces and objects in real-time and visualize them in 3D point clouds. In their paper, they extracted color and depth from RGB-D images and used them to reconstruct indoor spaces in real-time. Additionally, Choi et al. [11] proposed a 3D reconstruction pipeline that improves accuracy. Their method effectively handles errors that occur during the alignment of consecutive point cloud frames and automatically removes them, increasing the reconstruction accuracy even in cases with significant errors.

2.2 3D Object Classification & Detection

Indoor space objects composed of point clouds are subject to various external factors, such indoor spaces complexity, noise from RGB-D cameras, and reflective materials, resulting in significant noise and areas of information loss. Moreover, the post-processing approach using traditional 3D authoring tools can be cumbersome. Therefore, there arises a necessity for streamlining the remodeling process. To address this, ongoing research is focused on automating the classification of both spaces and objects. Furthermore, with the advancements in deep learning technology, there have been studies on classifying 3D objects using deep learning models. Notably, Qi et al. [5] proposed the PointNet neural network, which automates the classification of spaces and objects in indoor point clouds. PointNet utilizes the max-pooling function to extract global features of the point cloud. However, owing to the inherent characteristics of the max-pooling function, local information, excluding the maximum values,

is lost, which can lead to performance degradation in classification processes that require precise boundary delineation.

To address this issue, the PointNet++ neural network proposed by Qi et al. extracts feature vectors containing local information, improving the accuracy compared to the PointNet model [6]. PointNet++ utilizes a hierarchical U-Net architecture to extract local features of various sizes, integrating them to enhance classification and segmentation performance. In addition, in the VoteNet deep learning model proposed by Qi et al., the PointNet++ neural network is employed to design a network that detects 3D objects from point clouds [3]. By estimating the centroid of objects from the point cloud, points close to the object center are obtained, and these points estimate the 3D bounding box of the object based on the point features of the cluster.

2.3 Inpainting of the Lost Area

When using an RGB-D camera to capture indoor spaces and construct virtual environments, areas occluded by objects or not captured during the recording process result in information loss during the extraction of point clouds. Additionally, when classifying and removing objects, the corresponding point clouds of those objects are lost. Therefore, there is ongoing research on restoring the lost areas. Interpolation- and GAN-based methods are the two major methodologies for reconstructing missing regions in point clouds. Cui et al. [7] determined the boundaries of lost areas by clustering neighboring points and connecting them. Based on this, interpolation methods are employed to recover incomplete boundaries. The proposed approaches include reinstating lost areas through the utilization of the Poisson-based surface reconstruction algorithm. This involves leveraging detected boundary characteristics and using surface information symmetric to the boundaries. However, these methods may be unreliable when lost areas are large or numerous due to threshold-based boundary determination and inpainting.

Tchapmi et al. [8] proposed the TopNet neural network, which introduces a decoder that generates structured point clouds without assuming a specific structure or topology, generating point clouds following a hierarchical root tree structure. Zitian et al. [9] proposed the Pf-net neural network, which estimates the lost point cloud through a hierarchical generation network, which incorporates hierarchical completion and adversarial losses to generate missing areas. These existing studies contribute to the inpainting of lost point clouds, but obtaining the original data from noisy indoor point clouds can be challenging. In this study, we employ a inpainting method proposed by Ulyanov et al. [12], which allows inpainting without the original data by defining the lost areas using binary masks and restoring them by leveraging the correlations with non-zero pixels. It obtains promising results by learning the intrinsic image texture through pixel comparisons without directly considering the binary mask regions during inpainting.

3. Methodology

We employ RGB-D cameras to acquire images, subsequently separating them into color and depth images. This process enables us to derive the intrinsic structural details of the environment. Building upon this foundation, we proceed to reconstruct the virtual space. We estimate the structure of the constructed space, recognize 3D objects in it, remove the recognized objects, and restore the lost areas. **Fig. 1** shows the proposed pipeline.

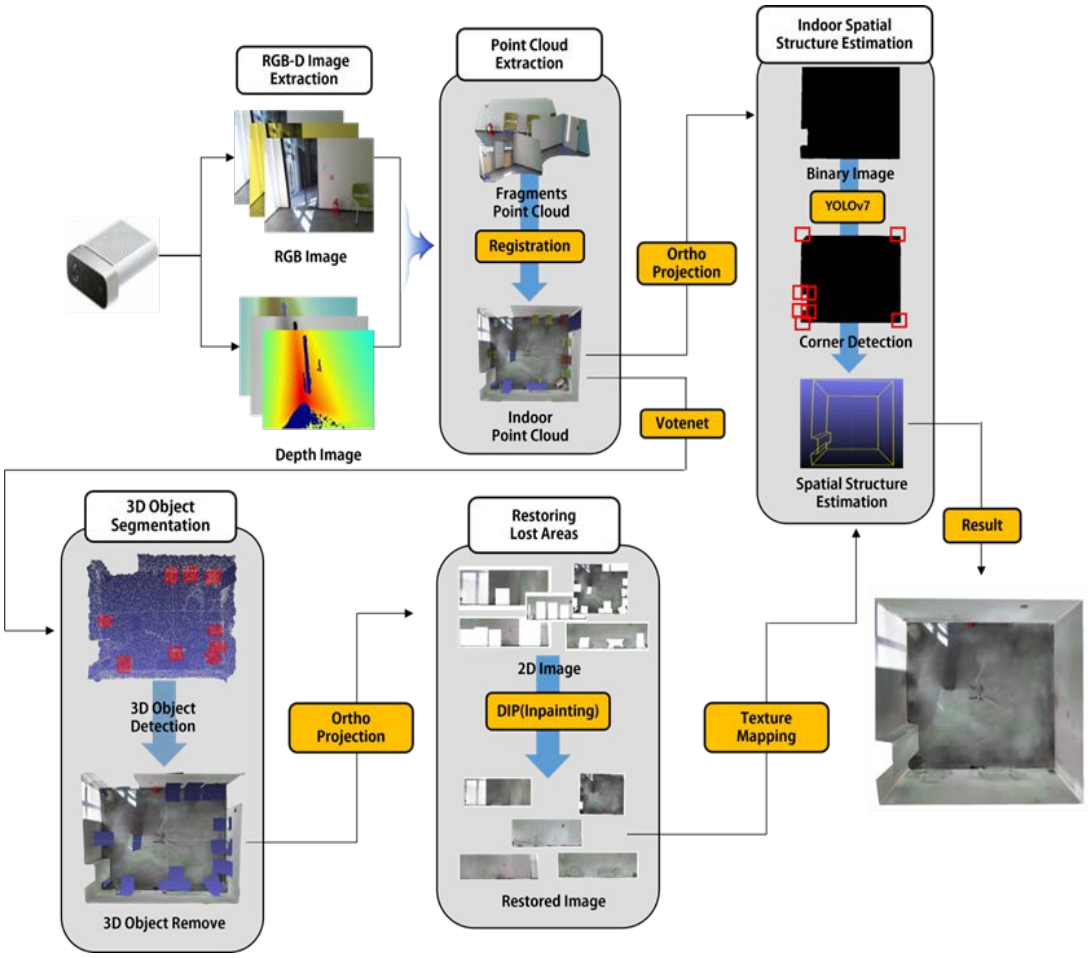

Fig. 1. Pipeline of the proposed spatial structure-based approach.

3.1 Indoor Point Cloud extraction

The Azure Kinect camera was used to acquire indoor spaces extract color and depth images, and the Open3D library was used to obtain the point clouds. The transformation relationship between RGB and depth image pairs was estimated using the ORB algorithm [13]. This involved identifying feature points across the image pairs and subsequently removing any incorrectly matched points. Furthermore, we deduced the camera trajectory by employing the 5-Point RANSAC algorithm [14].

Acquiring images using an RGB-D camera can introduce noise, and the accumulated errors make estimating the camera's position challenging. To address this issue, it is necessary to find consistency between the current frame and the registered camera positions in the past pose graph to reduce accumulated errors. Hence, we optimized the positions of the registered cameras in the pose graph to minimize errors.

$$x^{*}=argmin_{x} \sum_{ij} e_{ij}^{T} \Omega_{ij} e_{ij} \tag{1}$$

where x_i and x_j represent the position information of nodes in the current pose graph, the difference between the next frame node position and the previous frame node position is

denoted as $e_{ij}(x_i,x_j)$, Ω_{ij} represents the information matrix between the i_{th} and j_{th} nodes, and $argmin_x$ extracts the node with the smallest sum of the loss function.

Using this process, we used 5,300 color and depth images as input and divided them into 100 frames to extract 53 partial point clouds. As illustrated in **Fig. 2**, when inputted with the color and depth images, the corresponding partial point cloud is extracted.

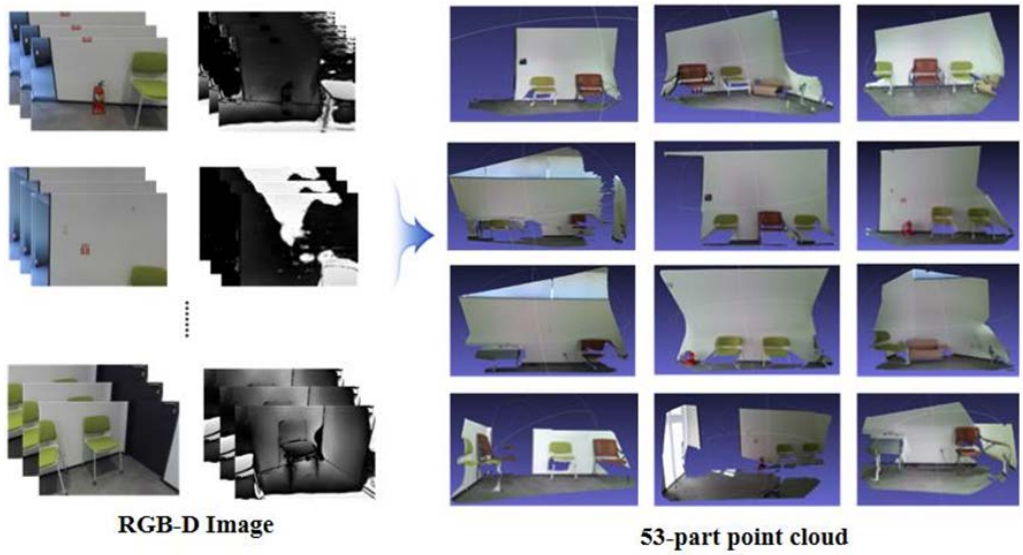

Fig. 2. Results of 53 partial point cloud extraction.

The Colored-ICP algorithm [15] was used to detect the correspondence relationships between the extracted partial point clouds. By aligning the partial point clouds, we obtained the complete indoor space point cloud, as illustrated in **Fig. 3**.

Fig. 3. Result of total indoor space point cloud extraction.

3.2 Deep Learning-Based Indoor Corner Detection

Although we obtained the complete indoor space point cloud, there are loss and noise areas, illustrated in **Fig. 4**, due to occlusion by objects and errors during the alignment of the partial point clouds. To address these issues, the space structure was estimated by extracting the floor plane by orthogonally projecting the complete indoor space point cloud. Thereafter, using the extracted floor plane, we detected the corners by applying the deep learning-based YOLOv7 [16] network rather than traditional computer vision-based corner detection methods such as Harris Corner, SIFT [18], and ORB [13], which are not suitable for point clouds owing to their irregular shape and noise contained in pixel values.

Fig. 4. Example of Missing areas and noise.

We converted the extracted floor planes into binary images and collected image data by restoring the lost areas. The collected data was classified into "Corner" and "Line" classes, and data augmentation techniques were applied to construct a total of 2,000 training data. **Fig. 5** provides an example of the constructed dataset.

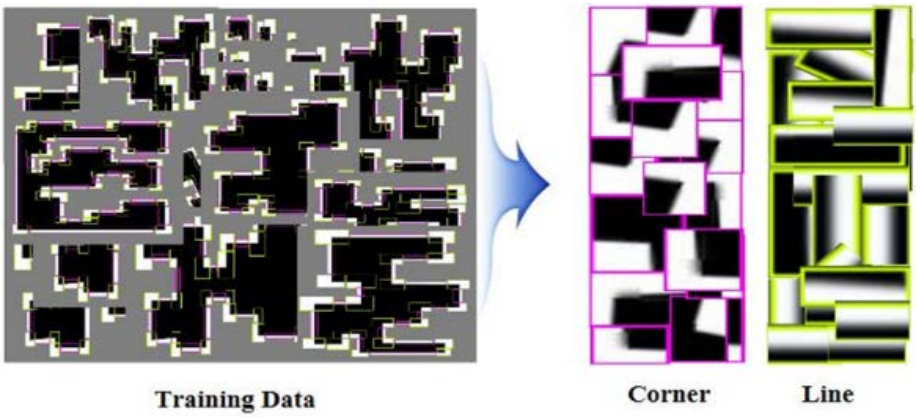

Fig. 5. Example of the corner & line train dataset.

Based on the constructed dataset, we trained the YOLOv7 using an Intel(R) Core (TM) i7-10700K CPU @ 3.80GHz 3.79 GHz processor and NVIDIA GeForce RTX 3090 GPU. The dataset consisted of 1300, 400, and 300 samples for training, validation, and test, respectively. We conducted the training for 300 epochs using the Adam optimizer with a learning rate of 0.001. **Fig. 6** shows the results of corner detection using the trained model.

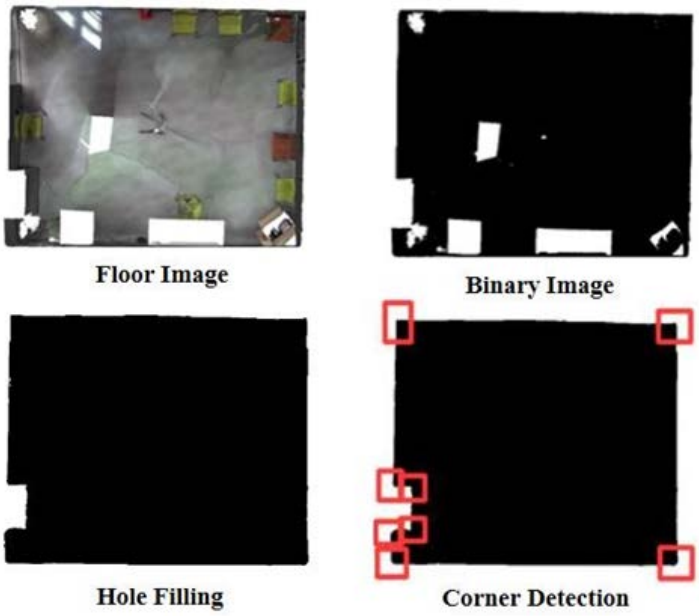

Fig. 6. Results of corner detection.

After corner detection, we visualized the 3D spatial structure by connecting bounding boxes' center coordinates and adding the z-coordinate to each point based on the constructed floor plane, as illustrated in. **Fig. 7**.

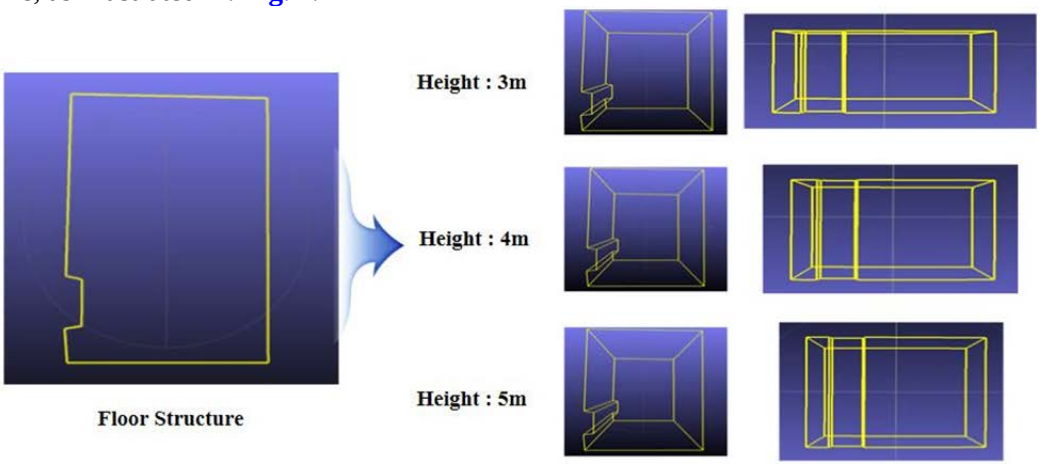

Fig. 7. Visualization of spatial structure.

3.3 3D Object detection

For 3D object detection, we employed VoteNet [3] using the SUN RGB-D dataset—a comprehensive repository of indoor point cloud data. It consists of 10,335 RGB-D images, 146,617 2D polygons, and 64,595 3D bounding boxes captured using Intel RealSense, Asus Xtion, Kinect v1, and Kinect v2 cameras. we extracted the data acquired with Kinect v2 from the SUN RGB-D dataset and trained VoteNet [3]. However, because our dataset point cloud was extracted based on images acquired with Azure Kinect, the coordinate system used during model training did not match, resulting in incorrect object detection, as illustrated **Fig. 8**.

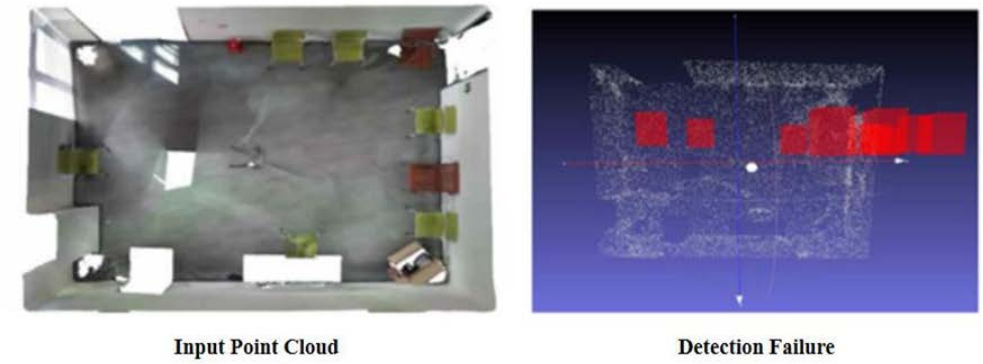

Fig. 8. Visualization of spatial structure.

we aligned the coordinate systems to perform object detection correctly. The point cloud extracted in this study had the y-axis pointing upward, while the point cloud used as training data had the z-axis pointing upward. Therefore, we rotated our point cloud by 90° along the x-axis to align the coordinate systems and reconducted object detection using the trained model. After object detection, we used the bounding box of the detected objects to apply object removal to the point cloud. **Fig. 9** shows the results after removing the detected objects.

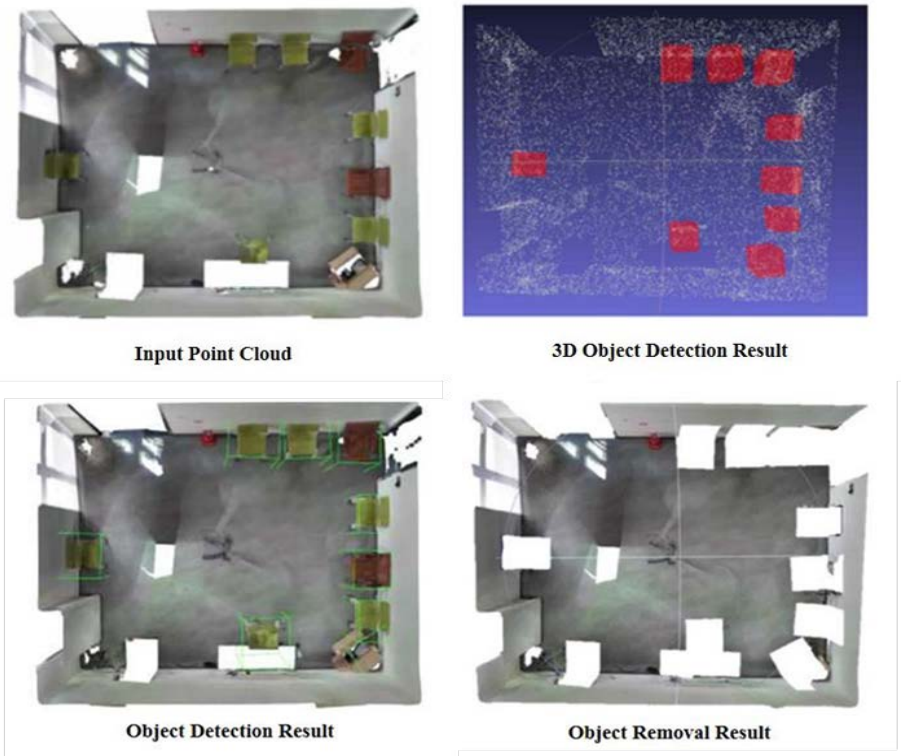

Fig. 9. Results of object detection removal.

3.4 Inpainting of Missing Area

After detecting and removing objects from the point cloud, we used an image-based inpainting approach to restore the missing and occluded areas where objects were located. The entire indoor point cloud was orthographically projected into a 2D image, and deep image prior (DIP) [12] was utilized for inpainting. To use DIP, the normal vector of each plane composing the indoor point cloud was extracted, and the binary image of the missing area was defined based on the orthographic projection of these normal vectors. **Fig. 10** shows the results of the orthographically projected planes and the corresponding defined missing areas. Subsequently, the missing areas were defined as binary images and fed as input to DIP, where the correlation among non-zero pixels was learned to restore the missing areas and perform inpainting, as illustrated in **Fig. 11**.

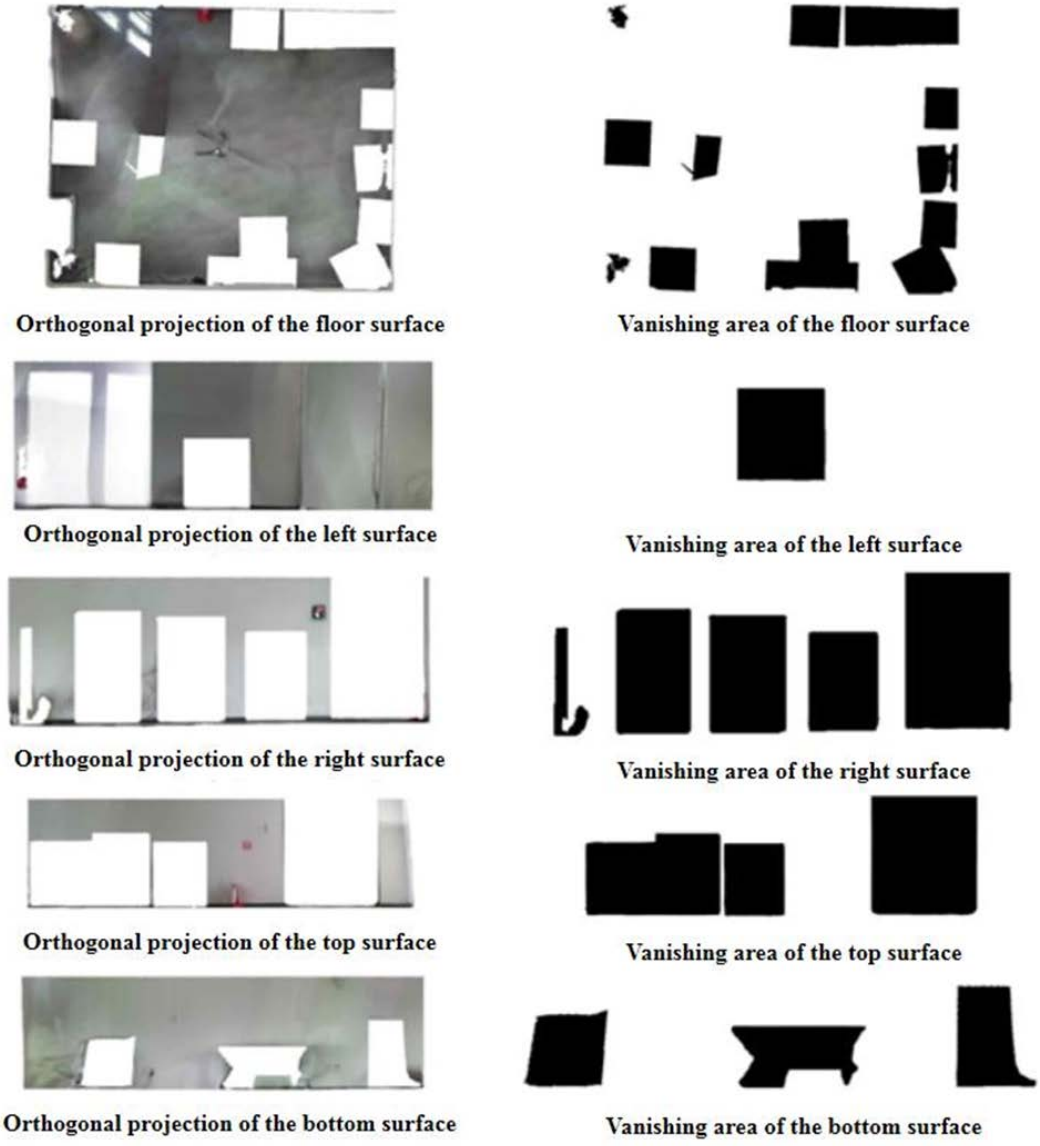

Fig. 10. Regions lost after orthogonal projection.

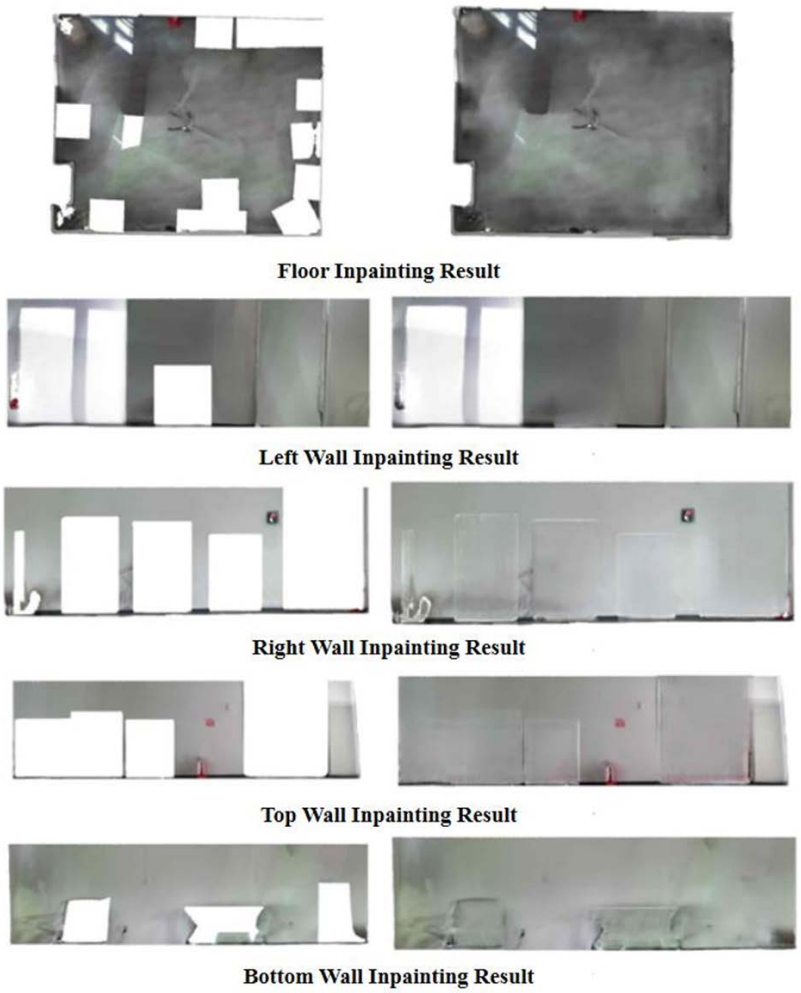

Fig. 11. Texture generation with DIP neural networks.

3.5 3D Model Face Generation and Texture Mapping

The estimated spatial structure consists of points represented in x, y, and z coordinates, interconnected to form surfaces. However, information pertaining to the orientation of faces is required to map textures onto these surfaces. As our estimated spatial structure only contains information about points, it is necessary to add information about the faces by selecting a minimum of three points. Thus, information for face construction was added using the spatial structure saved as an OBJ file. The OBJ file consists of *v*, *vn*, *vt*, *f*, *mtlilb*, and *usemtl*, where *v* represents 4-dimensional vertex information, *vn* represents 3-dimensional vertex normals, *vt* represents 3-dimensional texture information and coordinates, *f* represents information about the faces, *mtlilb* contains definitions for materials and textures, and *usemtl* indicates the specific texture to be used from the *mtl* file. We defined the coordinates of points sequentially, starting from the floor surface and subsequently progressing to the top, right, bottom, and left wall surfaces. This approach ensured that the point indexes remained properly aligned. **Fig. 12** displays the results of generating faces by aligning the coordinates for each wall surface.

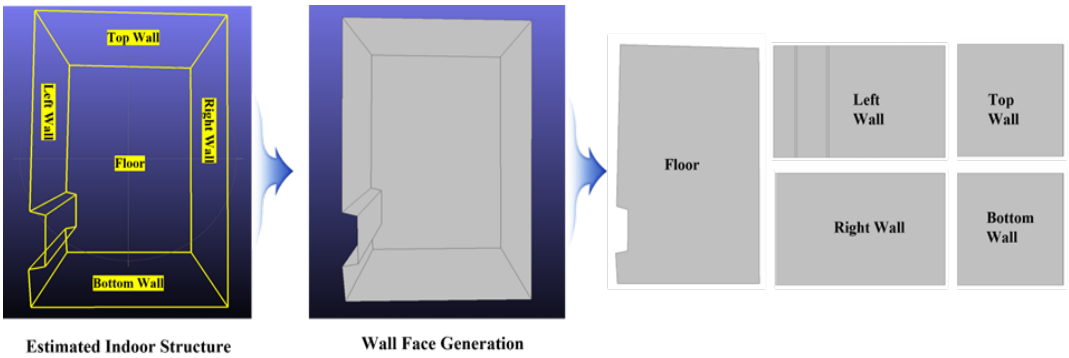

Fig. 12. Results of generating faces by aligning the coordinates for each wall surface.

After generating the faces, the restored textures, comprising 2D images, were applied to construct the 3D virtual indoor space. To apply the textures onto the generated surfaces, a transformation was necessary, and UV coordinates were used as the reference coordinate system for this transformation. The UV mapping technique was specifically employed to achieve this objective. UV maps were extracted using the trimesh API [17], and the textures were mapped based on them to construct the virtual indoor space. **Fig. 13** shows the UV map extraction and texture mapping results.

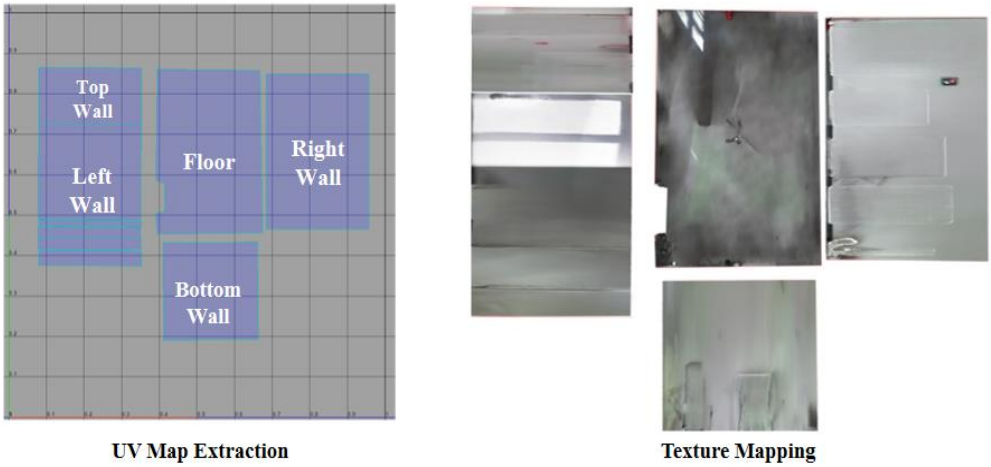

Fig. 13. UV map extraction and texture mapping.

4. Experimental Results

4.1 Corner Detection Results

The YOLOv7 was trained to detect corners on the floor, and the trained model was used for corner detection. To quantitatively evaluate the trained model, the metrics F1-Score, precision, and recall were used. The corner detection model performance is listed in **Table 1**.

Table 1. Corner detection model quantitative evaluation results.

Precision	Recall	F1-Score
0.9	0.85	0.87

We conducted a comparison between an existing corner detection algorithm, Harris corner detection [19], and our trained model. The Harris corner detection showed inaccurate results by detecting irregular pixel values on the image boundaries as corners, while our model detected corners more accurately, as illustrated in **Fig. 14**.

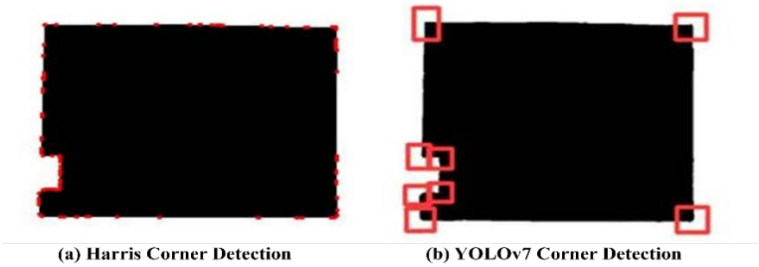

Fig. 14. Comparison of corner detection. (a) Harris corner detection, (b) YOLOv7 corner detection.

Fig. 15. Area restored with our proposed pipeline.

4.2 Qualitative Results

We compared the virtual spaces constructed using the proposed pipeline with those constructed using RGB-D camera-acquired images. When constructing the virtual space with

an RGB-D camera, there were noise and missing areas due to objects occluding the wall surfaces. However, by using the proposed pipeline, the wall surfaces of the virtual space showed significant improvement, with noise and missing areas effectively resolved. **Fig. 15** shows a comparison of the original face texture and the face texture created by our pipeline, and **Fig. 16** shows the outcome of using the mapped textures to construct the 3D virtual indoor space.

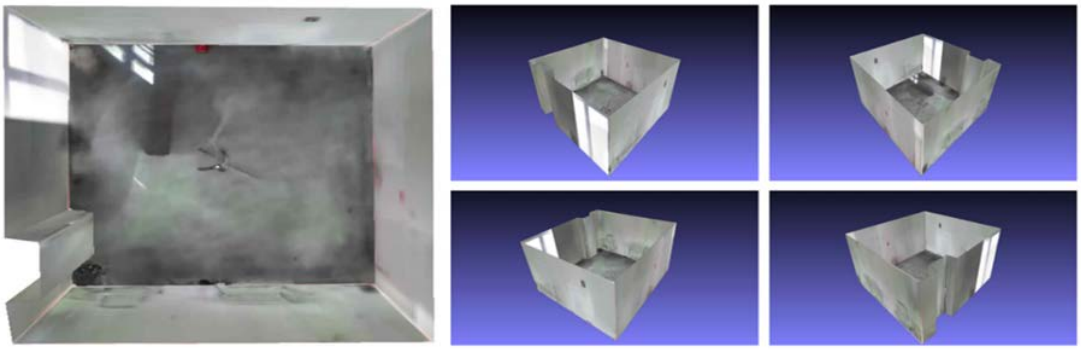

Fig. 16. Virtual indoor space construction

4.3 Extension for Application

Using the proposed pipeline, the constructed virtual space can be further edited and utilized with higher scalability based on the virtual space constructed using 3D modeling software. In this study, we integrated the objects modeled in Autodesk Maya into the constructed virtual space and rendered the final indoor virtual space. **Fig. 17** shows the virtual indoor space rendering result, incorporating both the constructed virtual space and objects modeled in Autodesk Maya.

Fig. 17. Virtual indoor space rendering result, incorporating the constructed virtual space and objects modeled in Autodesk Maya.

5. Conclusion

In this paper, we proposed a space structure-based modeling pipeline to efficiently create virtual indoor spaces while minimizing manual efforts and overcoming issues with traditional virtual space creation methods. The pipeline utilizes point clouds extracted from real-world indoor spaces acquired through RGB-D cameras as the base dataset. The proposed pipeline operates through indoor virtual space construction, indoor space structure estimation, 3D objects detection, texture generation, and texture mapping.

We reconstructed the virtual indoor space using 3D reconstruction techniques based on the Open3D library and addressed the noise in the extracted point clouds through space structure-based modeling. Additionally, we employed the VoteNet neural network for object detection and used the DIP neural network to restore the missing areas after removing the classified objects. As a result, we successfully constructed a virtual indoor space without objects, demonstrating its usability in 3D authoring tools and validating its practicality in VR, AR, and metaverse content creation. The proposed method allows users to construct and edit virtual spaces easily, significantly contributing to the entertainment industry.

Currently, our pipeline generates textures based on the outer walls of indoor spaces and maps them to the estimated space structure for constructing the virtual indoor space. However, wall thickness information is required to address the inner walls of indoor spaces. Therefore, future research should focus on generating thickness information and addressing the outer and inner wall textures, thereby further enhancing the construction of virtual indoor spaces.

Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00970, Development of multi-image-based 3D virtual space and dynamic object reconstruction technology for manufacturing site support, 50%) and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2022R1A2C1004657, 50%).

References

[1] C.W. Chu, J.Y. Park, H.W. Kim, J.C. Park, S.J. Lim and B.K. Koo, “Recent Trends of 3D Reconstruction Technology,” *Electronics and Telecommunications Research Institute on Electronics and Telecommunications Trends*, vol.22, no.4, Aug. 2007. Article (CrossRef Link)

[2] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, and E. Gobbetti, “State-of-the-art in Automatic 3D Reconstruction of Structured Indoor Environments,” *Computer Graphics Forum*, vol.39, no.2, pp.667-699, Jul. 2020. Article (CrossRef Link)

[3] C. R. Qi, O. Litany, K. He, and L. Guibas, “Deep Hough Voting for 3D Object Detection in Point Clouds,” in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.9276-9285, 2019. Article (CrossRef Link)

[4] T. Czerniawski, B. Sankaran, M. Nahangi, C. Haas and F. Leite, “6D DBSCAN-based segmentation of building point clouds for planar object classification,” *Automation in Construction*, vol.88, pp.44-58, Apr. 2018. Article (CrossRef Link)

[5] R. Qi Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.77-85, 2017. Article (CrossRef Link)

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Proc. of NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp.5105-5114, Dec. 2017. Article (CrossRef Link)
[7] L. Cui, G. Zhang, and J. Wang, "Hole Repairing Algorithm for 3D Point Cloud Model of Symmetrical Objects Grasped by the Manipulator," *Sensors*, vol.21, no.22, Nov. 2021. Article (CrossRef Link)
[8] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural Point Cloud Decoder," in *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.383-392, 2019. Article (CrossRef Link)
[9] Z. Huang, Y. Yu, J. Xu, F. Ni and X. Le, "PF-Net: Point Fractal Network for 3D Point Cloud Completion," in *Proc. of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.7659-7667, 2020. Article (CrossRef Link)
[10] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. of 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp.127-136, 2011. Article (CrossRef Link)
[11] S. Choi, Q.-Y. Zhou and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5556-5565, 2015. Article (CrossRef Link)
[12] V. Lempitsky, A. Vedaldi and D. Ulyanov, "Deep Image Prior," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.9446-9454, 2018. Article (CrossRef Link)
[13] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. of 2011 International Conference on Computer Vision*, pp.2564-2571, 2011. Article (CrossRef Link)
[14] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.26, no.6, pp.756-770, Jun. 2004. Article (CrossRef Link)
[15] J. Park, Q.-Y. Zhou and V. Koltun, "Colored Point Cloud Registration Revisited," in *Proc. of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp.143-152, 2017. Article (CrossRef Link)
[16] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in *Proc. of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.7464-7475, 2023. Article (CrossRef Link)
[17] Dawson-Haggerty et al., Trimesh, 2019. https://github.com/mikedh/trimesh
[18] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol.60, pp.91-110, Nov. 2004. Article (CrossRef Link)
[19] C. Harris, and M. Stephens, "A Combined Corner and Edge Detector," in *Proc. of the Alvey Vision Conference*, pp.147-151, 1988. Article (CrossRef Link)

Wonseop Shin received the B.S. degree in Computer engineering from Sungkyul University in South Korea from 2017 to 2022. He is currently pursuing a M.S. degree from Chung-Ang University, Graduate School of Advanced Imaging Science in South Korea since 2023. His areas of interest include Pose Estimation, Object Detection, Generative Models, as well as Virtual and Augmented Reality.

Jaeseok Yoo received the M.S. degree in Chung-Ang University, Graduate School of Advanced Imaging Science in South Korea from 2021 to 2023. He is currently working on the field of Computer Vision in NextChip. His areas of interest include Computer Vision, Deep Learning, ADAS, Autonomous Driving.

Bumsoo Kim received the B.S. degree in Art and Technology from Chung-Ang University in 2023, South Korea. He was an AI Researcher with the VIVE STUDIOS, South Korea. He is currently pursuing a M.S. degree in Applied Art and Technology at Chung-Ang University. His research interests include style transfer, face stylization/cartoonization, face re-aging and face swap.

Yonghoon Jung received the B.S. degree in computer engineering from Sungkyul University in 2022. He is currently pursuing a M.S. degree at The Graduate School of Advanced Imaging Science, Multimedia & Film at Chung-Ang University, specializing in Entertainment Technology. His research is centered around artificial intelligence and computer vision, with a particular focus on synthetic data generation and domain adaptation techniques. He aims to apply these cutting-edge technologies to solve complex issues in the real world. His dedication to his field is evident as he continues to explore innovative solutions to enhance technological applications.

Muhammad Sajjad received the M.S. degree from the Department of Computer Science, College of Signals, National University of Sciences and Technology, Rawalpindi, Pakistan, in 2012, and the Ph.D. degree in digital contents from Sejong University, Seoul, South Korea, in 2015. He is currently working as an ERCIM Research Fellow at NTNU, Norway. He is also an Associate Professor with the Department of Computer Science, Islamia College Peshawar University, Pakistan. He is also the Head of the Digital Image Processing Laboratory, Islamia College University Peshawar, where many students are involved in different research projects under his supervision, such as big data analytics, medical image analysis, multi-modal data mining and summarization, image/video prioritization and ranking, fog computing, the Internet of Things, autonomous navigation, and video analytics. He has published more than 65 papers in peer-reviewed international journals and conferences. His primary research interests include computer vision, image understanding, pattern recognition, robotic vision, and multimedia applications, with current emphasis on economical hardware and deep learning, video scene understanding, activity analysis, fog computing, the Internet of Things, and real-time tracking. He is serving as a professional reviewer for various well-reputed journals and conferences.

Youngsup Park received the B.S. degree in computer science and engineering from Daejeon University, Daejeon, South Korea, in 1995. the M.S. degree from the GSAIM Department, Chung-Ang University, Seoul, South Korea, in 2001 and Ph.D. degrees from the computer science and engineering from Chung-Ang University, in 2007. He was a Postdoctoral Researcher with Chung-Ang University, in 2007, and the Sungkyunkwan University, in 2008. He was a Technical Director with AR VISION Corporation from 2009 to 2015. He was a Director with Inno-Simulation Corporation from 2016 to 2024. His research interests include Spatial Computing, VR/XR. He is currently being evaluated as a reviewer for VR/spatial computing at IITP, KOCCA and KEIT.

Sanghyun Seo received the B.S. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 1998, and the M.S. and Ph.D. degrees from the GSAIM Department, Chung-Ang University, in 2000 and 2010, respectively. He was a Senior Researcher with G-Inno Systems, from 2002 to 2005. He was a Postdoctoral Researcher with Chung-Ang University, in 2010, and the LIRIS Laboratory, Lyon 1 University, from February 2011 to February 2013. He has worked at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, from May 2013 to February 2016. He has also worked at Sungkyul University, from March 2016 to February 2019. He is currently a Faculty Member with the College of Art and Technology, Chung-Ang University. His research interests include computer graphics, non-photorealistic rendering and animation, real-time rendering using GPU, VR/AR, and game technology. He has been a program committee member of many international conferences and workshops. He has been a Reviewer of Multimedia Tools and Applications (MTAP), Computers and Graphics (Elsevier), U.K., the Journal of Supercomputing (JOS), and The Visual Computer (Springer). He has edited a number of international journal special issues as a Guest Editor, such as the Journal of Real-Time Image Processing, the Journal of Internet Technology, and Multimedia Tools and Applications. He has been an Associate Editor of the Journal of Real-Time Image Processing, since 2017.