Mesh Enhancement of a 3D Volumetric Model using Generative AI for a Web 3.0-based Graphic Service

Byung-Seo Park[1], Ye-Won Jang[2], Hak-Bum Lee[2] and Young-Ho Seo[2,*]

[1]*HANTEO GLOBAL, INC., Korea*
[2]*Kwangwoon University, Korea*
E-mail: dev.bspark@hanteo.com; ywjanng@kw.ac.kr; hblee@kw.ac.kr; yhseo@kw.ac.kr
[*]*Corresponding Author*

Received 26 October 2024; Accepted 22 November 2024

Abstract

Using depth images from RGB-D cameras simplifies reconstructing 3D information for adaptive online transmission. However, depth sensors often produce distance-related distortions, leading to 3D distortions in reconstructed point clouds or meshes. This paper addresses these issues by proposing a method to enhance volumetric 3D data quality using synthesized point clouds and generating meshes with low-cost RGB-D cameras for Web 3.0 graphic services. We utilize calibration and reconstruction techniques from previous studies to create point clouds, refine them, and convert them into meshes. Finally, we improve the mesh surface using a latent diffusion model (LDM). The proposed calibration method reduced errors to 0.00926 mm in the 3D

Journal of Web Engineering, Vol. 24_1, 107–134.
doi: 10.13052/jwe1540-9589.2415

Charuco board experiment. For the Moai statue, the alignment accuracy achieved an average error of 8 mm and a standard deviation of 3.9 mm. Using LDM, the mesh surface improvement reduced the average error by 54.8% and the standard deviation by 65.9%.

Keywords: Web 3.0, graphic service, point cloud, 3D volume model, depth information, generative AI.

1 Introduction

Recently, RGB-D sensors, which combine RGB and depth sensors, have become widespread and have been utilized across various fields. Using RGB-D sensors allows for relatively accurate and rapid detection of the shape of captured objects. These sensors have significantly advanced a wide range of applications, including SLAM (simultaneous localization and mapping) and navigation [1,2], object tracking [3,4], object recognition and localization [5], pose estimation [6], and 3D volumetric model synthesis [7]. In an RGB-D sensor, color information is obtained through an RGB camera, while depth images are acquired through various types of sensors, such as laser distance scanners, time-of-flight (ToF) sensors, and structured light-based sensors [8].

To accurately and reliably reconstruct 3D volumes using an RGB-D sensor, it is necessary to compute the intrinsic parameters of each camera and the extrinsic parameters between the two sensors. In some cases, the parameters required for calibration are pre-provided in the form of a lookup table stored in the memory of the camera at the time of production. In real-time scanning and 3D volumetric imaging applications, where multiple RGB-D sensors are used, estimating the position and orientation between multiple cameras is a crucial challenge [9]. Extensive research has been conducted to obtain accurate intrinsic and extrinsic parameters of cameras. While there are various ways to classify calibration techniques, this paper categorizes and explains them based on depth estimation methods using structured light [10–13] and calibration techniques for depth estimation with structured light-based cameras [14–19].

First, several studies have focused on calibration techniques using structured light-based depth estimation methods. Khoshelham and Elberink [10] developed a calibration method for the Kinect sensor and provided an analysis of the accuracy and resolution of its depth data. They presented an analysis of the factors affecting this data. Data accuracy is based on a mathematical model of depth measurement derived from disparity. Mikhelson et al. [11]

proposed a method to estimate the position of corners in a point cloud derived from depth images to locate a checkerboard on a depth image plane. Staranowicz et al. [12] suggested a pose estimation algorithm for RGB-D sensors using a spherical object moving in front of the camera as a reference. Zheng et al. [13] developed an optical motion capture system capable of automatically estimating and correcting the camera's pose. This system comprises 12 cameras equipped with infrared LEDs surrounding the camera lenses, and it can track the positions of both the cameras and reflective markers in real-time.

Recently, there has been rapid advancement in low-cost structured light-based depth estimation sensors. Research on calibration techniques for these structured light-based sensors has also progressed. Jung et al. [14] proposed a method for calibrating the intrinsic and extrinsic parameters of a color camera and a time-of-flight (ToF) camera pair using a pattern with a 4 cm diameter hole that can be simultaneously recognized by both sensors. Mei and Rives [20] addressed the problem of finding the relative position between a 2D laser rangefinder and a refractive-reflective camera by using patterns recognized on a plane by both sensors. Scaramuzza et al. [21] proposed a method to map 3D distance information collected by a 3D tilting laser rangefinder onto 2D images that highlight key features of a scene. After manually linking points between the two sensors, they performed calibration of the extrinsic parameters using the perspective-n-point (PnP) algorithm and a non-linear refinement step. Perez-Yus et al. [17] presented a method for minimizing the overlapping area between images and estimating the relative pose between an RGB camera and a depth camera, offering flexibility in calibrating various sensors. Fukushima [18] proposed an ICP (iterative closest point)-based calibration method for ToF cameras.

We proposed an algorithm to perform 3D reconstruction by converting the input data from these cameras into a 3D point cloud after efficiently calibrating multiple RGB-D cameras [19]. While this method provides a robust calibration and reconstruction solution, it has limitations in removing outliers from the point cloud and generating meshes due to the inherent depth errors produced by RGB-D cameras. Therefore, this paper aims to present a solution to overcome these limitations by leveraging the rapidly advancing deep learning technologies.

Research in the field of generative AI has led to the development of text-to-image models, which enable high-quality image synthesis from text prompts, driving significant advancements in image generation. Generative AI is widely used across various fields and for diverse purposes. Among

these, diffusion models have demonstrated exceptional performance in image synthesis and ultra-high-resolution applications. Training models in this field, however, require substantial computational resources. To address this, samplers that support LDM, such as DDPM (denoising diffusion probabilistic model) [22], attempt to mitigate computational challenges by sampling a smaller amount of data in the early noise reduction stages. These models identify a more computationally efficient latent space while maintaining equivalent performance, allowing for the training of diffusion models tailored to high-resolution image synthesis.

Fine-tuning refers to further adjusting the parameters of a pre-trained model by training it on a new dataset, which is more efficient than training a model from scratch. In LDM, fine-tuning can be applied to the U-Net and the text encoder, and Dreambooth [23] supports fine-tuning in both areas. Additionally, Ben Mildenhall and colleagues proposed a new method for representing 3D volumetric models, known as NeRF(neural radiance fields) [24]. NeRF takes in 5D data to generate views of objects from new perspectives by creating functions that extract brightness and density from input data using deep learning algorithms.

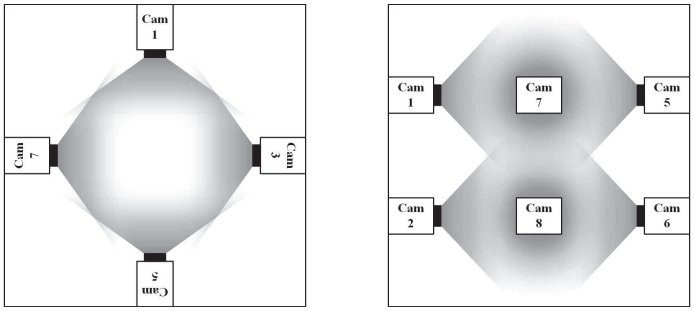
This paper is organized as follows. Section 2 briefly explains the camera systems and algorithms used in previously proposed methods and introduces the LDM employed in this study. Section 3 describes the proposed methods for point cloud refinement and mesh generation. In Section 4, we present the experimental results of the proposed approach, and in Section 5, we conclude the paper.

2 Related Works

In this section, we explain the multiview RGB-D camera system we use, the method for reconstructing 3D point clouds through this system, and the fundamental theory of the LDM used to improve the mesh quality generated from the 3D point cloud.

2.1 Camera System

In this section, we explain the configuration and characteristics of the multi-view RGB-D cameras. A multi-view RGB-D camera system is a setup in which multiple cameras are installed in a fixed space to scan an object from various perspectives. To generate 3D volumetric models, this study employs eight cameras to create the multi-view RGB-D camera system. Figure 1



(a) (b)

Figure 1 Multiview RGB-D camera system for scanning 3D volumetric models: (a) vertical shooting angle and range, (b) horizontal shooting angle and range.

illustrates the multi-view RGB-D camera system arrangement used in this research. As shown in Figure 1(a), the eight cameras are positioned facing the center of the space, and as depicted in Figure 1(b), four cameras are placed on the upper level and four on the lower level. Each camera consists of an RGB-D sensor pair.

The placement of the cameras is determined by considering the type and performance of the RGB-D sensors, as well as the size and distance of the target object to be synthesized. The quality of the synthesized 3D volumetric model and the frame rate per second depend on the characteristics of the RGB-D sensors. Typically, the number and type of RGB-D sensors are selected based on the intended application of the final 3D volumetric model. For this study, we chose the Kinect Azure [25], a structured-light-based depth sensor, as it offers a relatively low cost while providing suitable resolution for both RGB and depth images, sufficient output frame rates, and appropriate operational range for the task at hand.

In this paper, point clouds and meshes are generated using RGB-D cameras; however, it is equally feasible to use standard commercial RGB cameras. The core focus of the proposed algorithm is based on refining incomplete meshes, allowing flexibility in the choice of methods for generating point clouds or meshes. Among these various methods, our paper specifically employs multi-view RGB-D cameras.

2.2 3D Point Cloud Reconstruction

This section describes the process of generating the 3D point cloud used in this paper. This process is illustrated in Figure 2 and is broadly divided

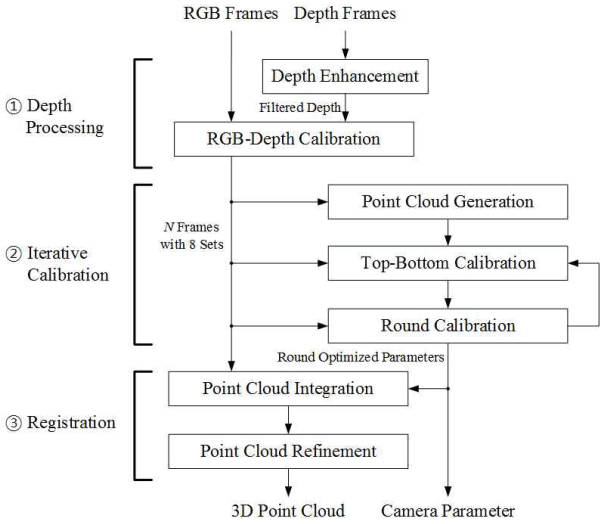


Figure 2 The proposed procedure of extrinsic calibration for generating the photorealistic 3D volumetric model using the multi-view RGB-D camera system.

into three stages: depth processing, iterative calibration, and registration [19]. In depth processing, since the depth images captured by low-cost structured light sensors contain significant noise, appropriate noise reduction techniques are applied to remove errors from the depth images. After noise removal, each camera's depth image is calibrated based on the corresponding RGB image. Next, the iterative calibration consists of two main steps. First, shared feature points between the cameras positioned at the top and bottom are identified, and this information is used to calculate the extrinsic parameters. Next, the extrinsic parameters are calculated by predicting the camera positions based on the feature point locations among the four viewpoints. This process is repeated iteratively until the errors converge. Finally, in registration, using the extrinsic parameters obtained from the iterative calibration process, all the point clouds are integrated. A refinement process is then performed on the point cloud to enhance its quality.

2.3 Latent Diffusion Model

The LDM operates within a latent space rather than the data space during diffusion. The reason for using latent diffusion instead of stable diffusion is that the number of values to process is significantly reduced. There are various methods for transitioning from data space to latent space, and in this

study, we use the most common approach: the variational autoencoder (VAE). In LDM, the VAE is trained to represent data in a lower-dimensional space. Unlike a traditional autoencoder, which focuses on compressing data and extracting features effectively, the VAE is a generative model that emphasizes decoding, aiming to generate new data. The VAE aims to generate new data by modeling the probability distribution of image data. When data passes through the encoder, the VAE produces two outputs: the mean and standard deviation, which can be used to create a normal distribution. Latent vectors sampled from this distribution pass through the decoder to generate data with a distribution similar to the input. However, because sampling introduces randomness, direct computation is impossible. The reparameterization trick addresses this, enabling sampling while maintaining computational feasibility. From the reduced complexity provided by the VAE, LDM can efficiently generate images by passing through the network only once in the latent space. Figure 4 illustrates the LDM architecture. A key advantage of this approach is that the VAE's encoding step only needs to be trained once, making it reusable across multiple diffusion models. This allows for the efficient application of various diffusion models tailored to different purposes.

3 Proposed Enhancement Method

This section explains the proposed method for enhancing the quality of the 3D point cloud and generating the mesh.

3.1 Overview

This paper proposes an external calibration technique for a multi-view imaging system equipped with multiple unaligned structured light-based RGB-D sensors and a method for synthesizing 3D volumetric models using generative AI. Our approach consists of the three major algorithmic stages defined in Figure 3. The first stage involves introducing an iterative calibration method for multi-view cameras to minimize parameter errors that arise during the calibration process. The second stage focuses on enhancing the calibration results and the quality of the synthesized point cloud by effectively mitigating depth distortion and noise, which are inherent to structured light cameras depending on distance, through filtering depth information. These two processes have been previously suggested in earlier studies [19]. The third stage aims to increase viewpoint continuity between cameras and accurately synthesize surface information by utilizing an AI learning model and training method specifically designed for this system.

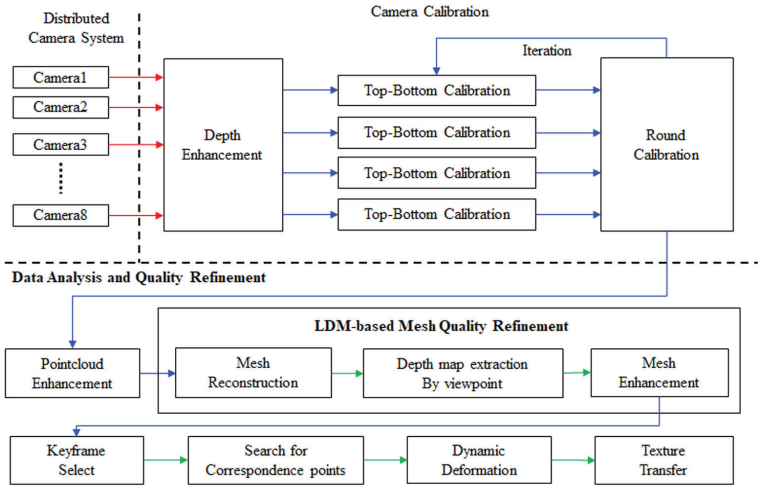


Figure 3 3D volumetric model reconstruction using generative AI.

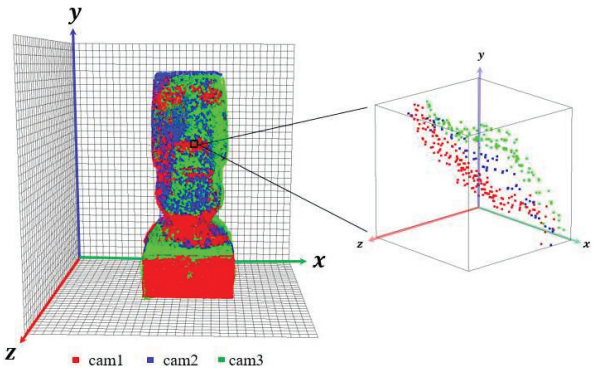


Figure 4 Data distribution by the camera inside the nth voxel.

The structured light cameras involve depth noise that increases proportionally with distance. The integrated point cloud generated from the object may contain significant noise depending on the relative positions of the cameras. Additionally, overlapping point clouds will result in varying density levels. This scenario is illustrated in Figure 4. The box depicted in Figure 4 represents a quantized space, a sampled voxel. Each voxel contains point clouds captured by different cameras, and these point clouds are stratified into multiple layers based on factors such as calibration quality, the distance between each camera and the subject, and environmental conditions like lighting.

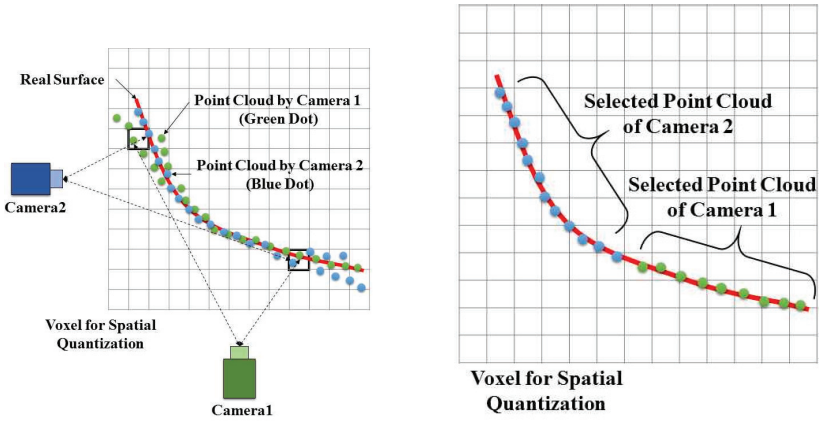
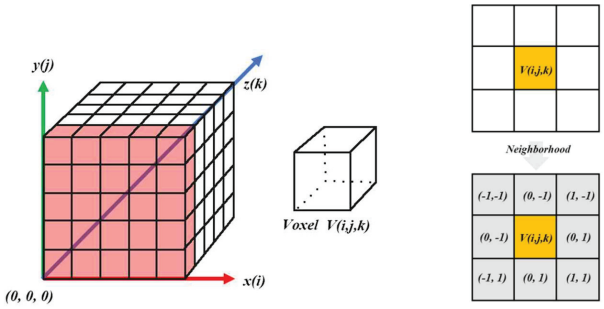


Figure 5 Method for selecting duplicate point clouds generated by multiple cameras (refinement algorithm). (a) Distribution of point clouds according to camera positions, (b) point clouds selected through the refinement algorithm.

In this paper, we propose a voxel-based neighborhood search method to reduce depth noise and adjust the spatial density of the point cloud. The algorithm is illustrated schematically in Figure 5 and builds upon our previous work [19]. As shown in the algorithm in Figure 5, the world coordinate system, which contains the integrated point cloud, is first divided into small three-dimensional grids, or voxels, and the point clouds from each camera are generated within this voxel space. Next, for each voxel, the object points obtained from the closest camera are selected from among the object points captured by each camera. This selection is made by calculating the Euclidean distance between the object points and the cameras. Since the depth images obtained from depth cameras are more accurate when the camera is closer to the object, the algorithm reflects this characteristic. Figure 5(a) illustrates the process of selecting point clouds by calculating the distance from the camera, while Figure 5(b) shows the point cloud selected through this process. In each voxel, all point clouds are grouped by camera, and only the points corresponding to the camera closest to the voxel position are retained.

Next, to address situations where the distances between each camera and the voxels are similar or where the influence of the selected camera is discontinuous across voxels, we propose an algorithm that ensures the continuous distribution of each camera's influence throughout the voxel space, as illustrated in Figure 6. This algorithm is based on a voxel-based neighborhood



(a) (b)

Figure 6 Coordinate system and neighborhood of a voxel. (a) voxel grid, (b) Neighborhood in the same layer.

search method, where the entire voxel space is searched, and clustering is performed to ensure that the influence of the selected camera at each voxel is smoothly distributed. The voxel search prioritizes the plane with the smallest index along the z-axis and repeats the search for the entire voxel resolution along the z-axis. Once the voxel plane to be searched is determined, eight neighboring voxels are selected from the current position. Among these eight neighboring voxels, the camera with the highest selection frequency is assigned to the current voxel, ensuring continuity between adjacent voxels.

3.2 Mesh Synthesis and Remeshing

To convert the 3D point cloud into a mesh, we use the Poisson disk sampling, Voronoi diagram, and Delaunay triangulation as illustrated in Figure 7. The blue squares represent the sample points generated by our method. The points are distributed evenly across the plane, ensuring that the points are spaced out by at least a minimum distance, which helps prevent clustering when creating the mesh. Each region contains all points closer to its defining sample point than any other. The dashed lines implicitly outline the regions around points A, B, C, and D, showing how the Voronoi cells would be divided. These cells are used as a preliminary step in creating the triangular mesh. Delaunay triangulation is the dual graph of the Voronoi diagram. It connects the sample points to maximize the minimum angle of the triangles, avoiding narrow triangles. The red points (A, B, C, and D) are the vertices of the triangles, and they are connected by solid lines, forming a triangular mesh. The edges of the triangles are drawn between adjacent points, ensuring that no point lies inside the circumcircle of any triangle. The triangular mesh is formed by

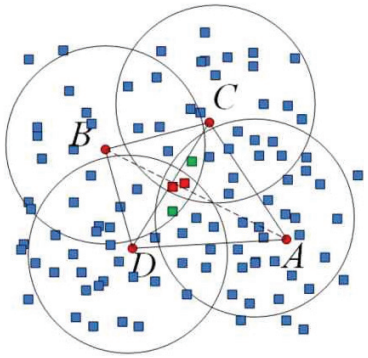
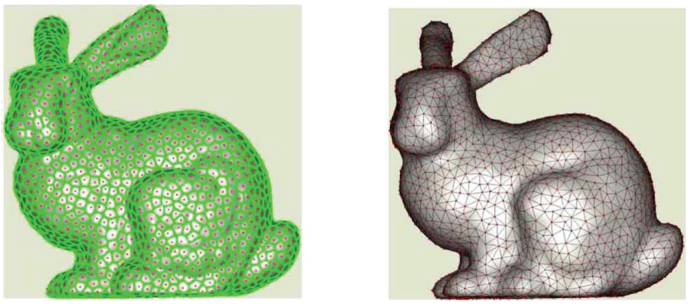


Figure 7 Triangle mesh construction method using Poisson disk sampling and a Voronoi diagram.



(a) (b)

Figure 8 (a) Voronoi diagram, (b) triangulated mesh structure.

connecting points A, B, C, and D, creating triangles that cover the area. The green squares in the center likely represent a specific region of interest, where the triangulation might be denser or under closer consideration. These small green squares indicate a refined area within the overall mesh structure.

As shown in Figure 8(a), a subset of data is sampled from the point cloud based on geodesic distances, and a Voronoi diagram is constructed. The process then involves synthesizing the triangular mesh by connecting the central axes of the Voronoi cells, as depicted in Figure 8(b). In the remeshing process, a combination of edge split, edge collapse, edge flip, and vertex shift is used to structurally align the surface consistently. The most crucial criterion set in this paper for improving surface quality is the minimum/maximum angles of the vertices that form the triangles. This is because acute triangle structures are advantageous for calculating the geodesic distances between

corresponding points for keyframe deformation [26–28]. If an acute angle smaller than the criterion appears or an obtuse angle more significant than the criterion arises, the angles of the triangles are equalized.

3.3 Mesh Quality Improvement Based on LDM

After all sequence configurations are completed, to enhance mesh quality, depth images from six directions (front, left, back, right, top, and bottom) of the 3D volumetric model are acquired from the camera perspective of a graphics pipeline that displays 3D data, such as OpenGL. These depth images are used as inputs for LDM. The inference process of LDM utilizes the depth images from each RGB-D sensor to train a U-Net, and the trained ControlNet weights, which condition the model based on the actual shape of the subject, are used together with these learned weights. For more efficient training, unique identifiers are assigned, allowing the features of high-quality depth images to be transferred through style transfer. Figure 9 shows the overall algorithm that enhances the quality of depth images obtained from the rendered 3D volumetric model through LDM. Each depth image acquired from the 3D volumetric model is converted into a latent vector of 1/8 the input resolution via a VAE. The latent vector derived from the depth images is conditioned through ControlNet. During the forward propagation stage, noise is progressively added at each step of the U-Net input as it passes through a sampler (DDPM [22], DDIM [29]). In the backward propagation stage, the amount of noise embedded in the latent vector is iteratively estimated and restored at each time step. The latent vector that has passed through

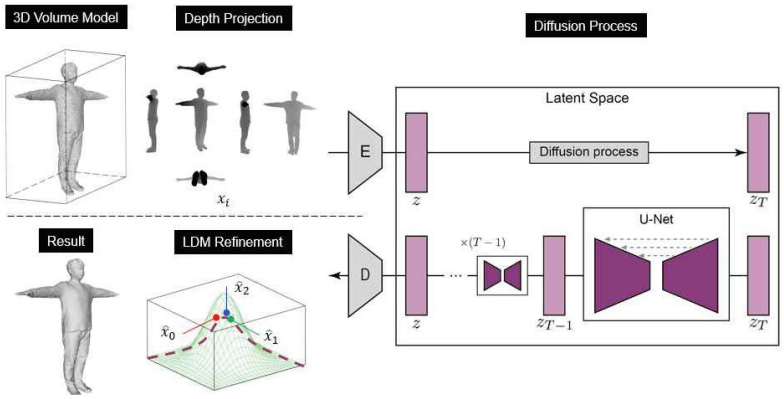


Figure 9 Mesh improvement algorithm using the LDM model.

the sampler is then reconverted to the image domain through the decoder of the VAE, at which point the enhanced depth images for each viewpoint are obtained. From the improved depth images acquired at each viewpoint, rays are generated toward the subject based on the depth value of each pixel, and the vertex positions on the mesh surface are adjusted by averaging the lengths of each ray.

4 Experimental Results

In this section, various experiments are conducted to verify the effectiveness of the algorithms proposed in Section 3, and the results are presented. First, the analysis of depth errors proposed in Section 3 and the results of the improvements made are shown, emphasizing the importance of the stages in enhancing the synthesis outcomes. Next, the results of the proposed calibration algorithm are verified through various experiments. In this paper, quantitative analysis is attempted using two datasets (3D Charuco board and Moai statue) that already have ground truth data. Following this, the results of the point cloud and mesh quality improvements proposed in Section 3 are presented, and the quality of the enhanced mesh is verified through an analysis of simulated noisy and distorted 3D data and actual volumetric models captured.

The experimental environment depicted in Figure 10 was constructed to conduct the proposed calibration and alignment experiments. The experimental setup shown in Figure 5 was built to analyze depth errors based on the shooting distance. This study used eight Azure Kinect RGB-D [25] cameras from Microsoft. The cameras were installed as described in Figure 1. As shown in Figure 10, four cameras were positioned at higher locations, and the other four were installed at lower locations, with each camera placed at heights of approximately 1 m and 2 m from the ground. A rail system capable of precise horizontal movement was built to measure the depth based on shooting distance, allowing for the accurate relocation of the cameras during the experiment.

4.1 Mesh Synthesis and Remeshing Results

Once the improvement process in the point cloud domain is completed, the point cloud is converted into a mesh to construct the surface. Additionally, a remeshing process is performed to enhance the surface structure. The results of this process are shown in Figure 11. Compared to the pre-remeshing

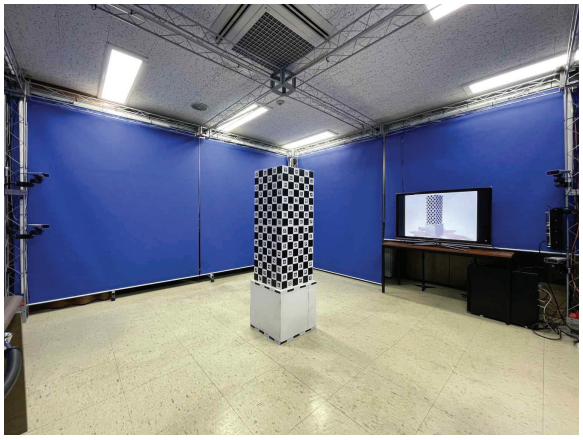
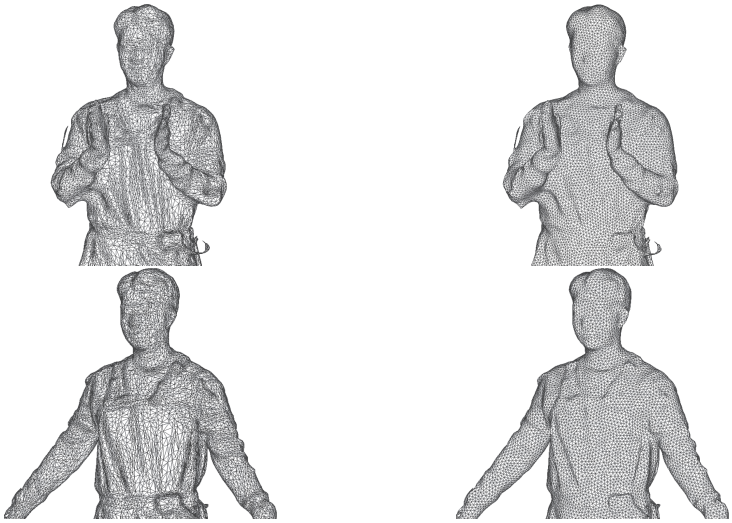


Figure 10 Built experimental environment for capturing the 3D object.



(a) (b)

Figure 11 Result of remeshing application, (a) before application, (b) after application.

results in Figure 11(a), the post-re-meshing results in Figure 11(b) exhibit a more uniform triangular mesh structure closer to equilateral triangles, demonstrating geometrical features with structural similarity.

Research on remeshing [30] or re-topology aims to reconstruct irregular surfaces into simple and regular surfaces. The definition of high-quality

surfaces includes fidelity, simplicity, and the quality of components [31]. This implies that to represent the geometric structure of the surface faithfully, the number of vertices and the complexity of connections should be reduced, and the mesh structure needs to be simplified. Furthermore, efficient computations on the surface require well-shaped triangular meshes [32]. Remeshing techniques modify the mesh structure of the input [33] or generate an entirely new mesh from scratch [34].

Structured remeshing replaces an unstructured input mesh with a structured mesh. A structured mesh consists of faces and edges where every internal vertex is connected by a consistent number of elements, allowing for more efficient traversal in algorithms.

4.2 LDM Training Results for Mesh Quality Improvement

To enhance the quality of the synthesized 3D volumetric model, the dataset used for LDM training was selected from InteriorNet [35]. It comprises 10,000 matched pairs of color and depth images chosen from the 20 million publicly available images. The InteriorNet dataset provides simulated pairs of color and depth images based on approximately 1 million CAD models supplied by furniture manufacturers. Each data sequence includes surface normals, camera motion paths, Kinect noise, lens distortion, and lighting conditions. The depth images simulated from actual 3D models serve as suitable ground truth data, as they are unaffected by occlusions, noise, or the quality of the shooting environment and sensors, making them closer to real depth images.

U-Net training in LDM was conducted using ControlNet to generate improved-quality depth images. In the preprocessing stage, Canny edge data was extracted from the color images, and the original depth images were converted into latent vectors through VAE, which were then used as the initial input for the DDIM [29] sampler. Each hyperparameter was fine-tuned through experimentation, with the initial learning rate set to 0.001 based on the experiments shown in Figure 12 and the learning rate scheduler set to Linear. The optimizer used was Adam. The training was carried out on a single Nvidia A100 (80G) GPU, and the training time until convergence was approximately 10 GPU hours. The graph illustrates the training tendency of an LDM under three learning rates: 0.00005 (blue), 0.0001 (orange), and 0.001 (gray), with the loss on the vertical axis and epochs on the horizontal axis. A learning rate of 0.00005 shows a steady and stable decrease in loss over time, indicating consistent training, while 0.0001 achieves faster

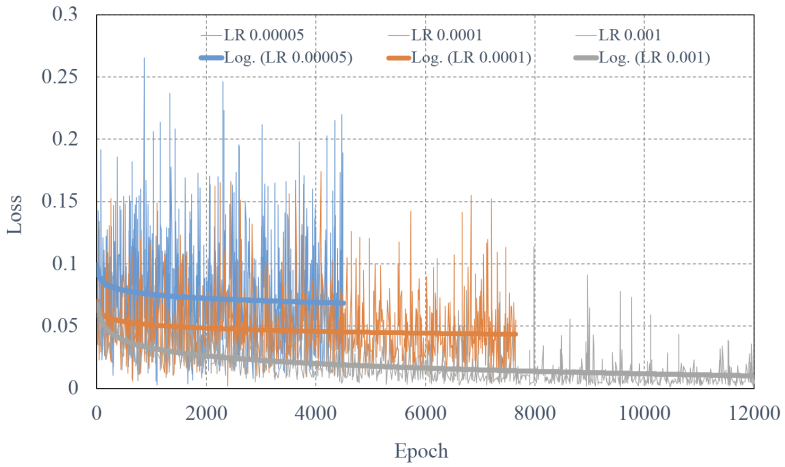


Figure 12 Training tendency by various learning rates in LDM.

initial loss reduction but exhibits greater fluctuations, suggesting a trade-off between speed and stability. In contrast, the 0.001 learning rate rapidly reduces loss initially but remains unstable throughout training, highlighting that it may be too high for effective convergence. The graph emphasizes the importance of selecting an optimal learning rate to balance convergence speed and stability.

In the inference stage, planar data at various distances, captured directly using Kinect Azure, and a separate dataset (Tsukuba, Venus, Teddy, Cone) not used in training were employed. The dataset used for LDM inference was downsampled to 25% of the original size, and the pixel size was then scaled back up four times to input degraded depth information. The experimental results shown in Figure 13 demonstrate the depth image quality improvement after applying LDM and 2D and 3D JBU [36, 37] methods to the Tsukuba, Venus, Teddy, and Cone datasets. The depth images processed by LDM show improved noise reduction and edge sharpness compared to the original 2D and 3D JBU methods in Figure 13.

NeRF (neural radiance fields) and the mesh quality improvement algorithm proposed in this paper using LDM share significant similarities in their approach of applying ray-based computations from the camera's viewpoint toward the object in projective space. The objective of NeRF is to generate novel views of an object, which it has not been trained on, by taking the camera position as input. However, this paper aims to synthesize enhanced 3D surface information using the camera viewpoint and object distance

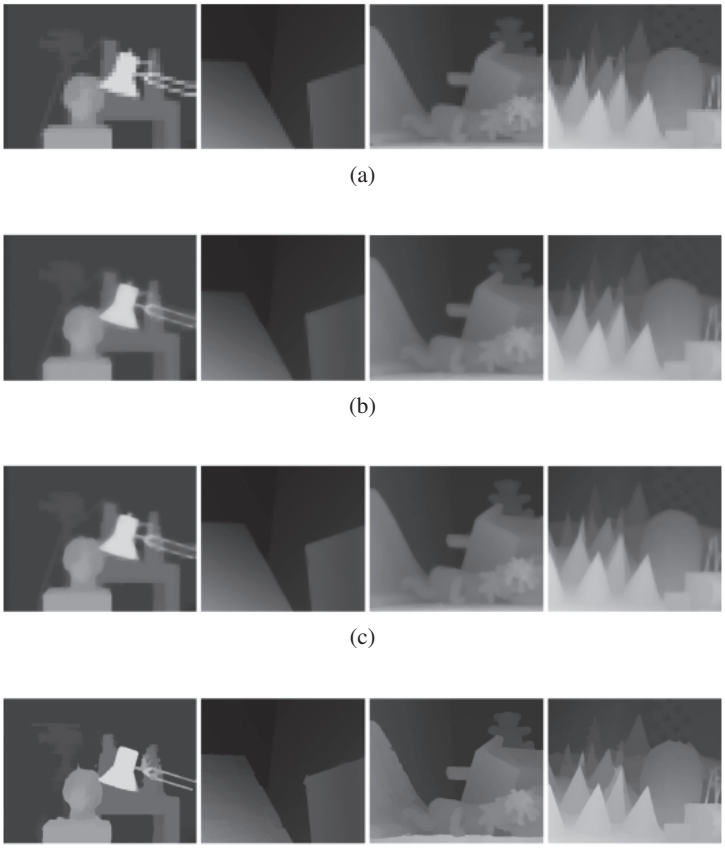


Figure 13 Comparison of depth information correction results. (a) Initial Depthmap, (b) 2D JBU, (c) 3D JBU, (d) proposed method (LDM).

information. To ensure versatility, it focuses on learning high-quality depth image features, allowing the application of a single set of weights to various 3D volumetric models. Table 1 compares the characteristics and performance of the NeRF method and the proposed method. While NeRF requires methods like triangulation and bundle adjustment to synthesize volumetric models, the approach proposed in this paper improves the quality of already synthesized 3D volumetric models through calibration and the synthesis methods of RGB-D sensors. Recently, NeRF has diversified into various forms, with different models tailored to specific applications. Since comparing and evaluating all

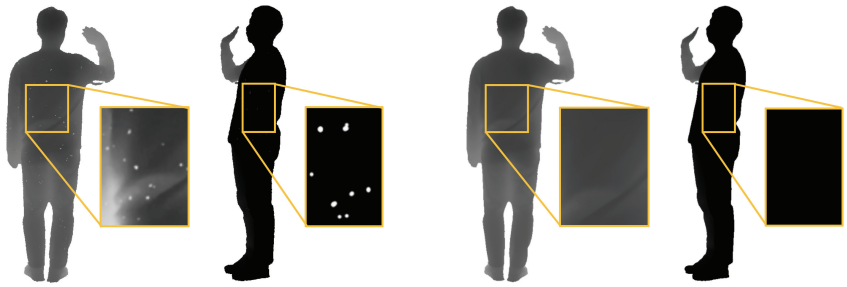
Table 1 Comparison table of the NeRF method and the proposed method

	NeRF (Vanilla)	Ours
Number of input data	100	10,000
Training time	1 day or more	10 hours
Versatility (number of models per object)	1 Model required per object	1
Dynamic reconstruction	×	○
Volume model reconstruction	×	○
Rendering time (1 frame inference time)	30 seconds	10–20 seconds

these diverse NeRF models is not feasible, the paper highlights the fundamental differences by comparing the vanilla NeRF model to the proposed method. Our method outperforms NeRF (Vanilla) in several key aspects, offering enhanced functionality and efficiency. While NeRF requires only 100 input data points, it needs more versatility and takes over a day to train, making it impractical for dynamic scenarios or real-time applications. Additionally, NeRF does not support dynamic or volumetric model reconstruction, further limiting its usability. In contrast, our approach leverages 10,000 input data points to achieve more detailed and accurate results while significantly reducing training time to 10 hours. It supports dynamic reconstruction and volumetric modeling, enabling broader applicability across various use cases. Moreover, our rendering time per frame (10–20 seconds) is faster than NeRF's 30 seconds, making it better suited for tasks requiring quicker output. However, the need for more input data in our method may increase data preparation efforts. Our method offers greater versatility, faster processing, and higher-quality outputs, particularly in dynamic and complex scenarios.

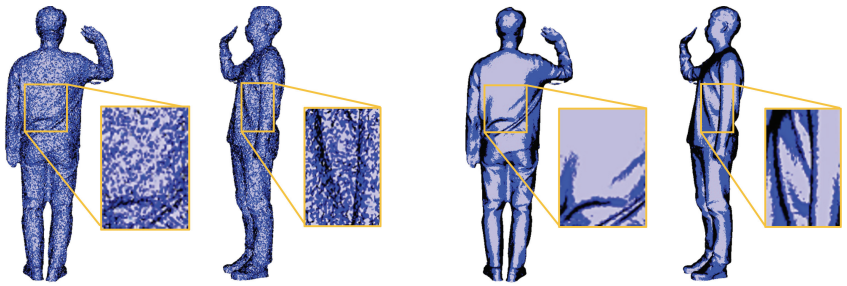
4.3 Mesh Quality Improvement Results

Figure 14 shows the depth images from different viewpoints of the 3D volumetric model improved through LDM. Figure 14(a) presents the depth images from four viewpoints. The noise in the depth information is visible in Figure 14(a), and the areas that were not successfully synthesized into the mesh due to this noise are shown to be restored and filled in Figure 14(b).



(a) (b)

Figure 14 Comparison of depth image correction images using LDM. (a) Original, (b) after improvement.



(a) (b)

Figure 15 Mesh surface information using depth image correction image using LDM. (a) Original, (b) after improvement (purple: 0°–30°, blue: 31°–60°, black: 61°–90°).

In Figure 14, the depth images from different viewpoints of the improved volumetric model are used to illustrate the structure of the enhanced mesh surface, shown in Figure 15. Each legend divides the surface normals into three stages based on angle to distinguish between noisy and improved surfaces. Figure 15 shows the image before applying LDM, while Figure 15 shows the results after applying LDM. Figure 15 shows that the surface undulations caused by noise, which are visible in Figure 15, have been smoothed out.

To quantitatively measure the performance of the mesh quality improvement algorithm using LDM, noise (-1 to 1 cm) was synthesized into the Stanford Rabbit and Armadillo datasets. The depth image enhancement and mesh improvement algorithms were then applied to each viewpoint, and the error compared to the ground truth was measured. The results are shown in Figures 16 and 17.

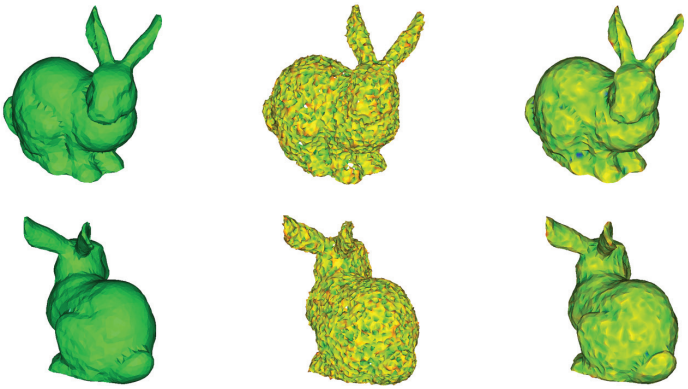


Figure 16 Stanford Rabbit noise synthesis and improvement results. Left: original, center: noise synthesis model, right: improved model using LDM.

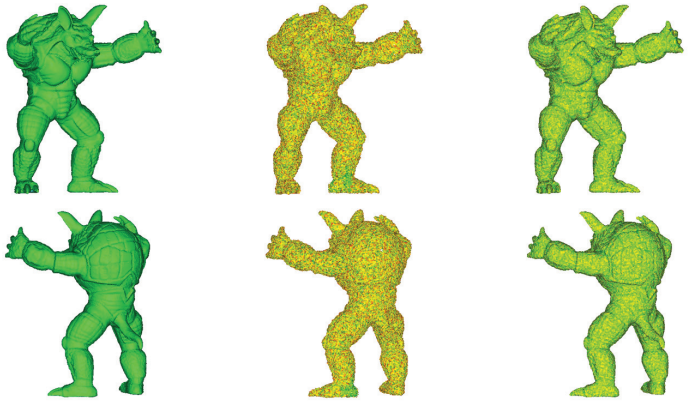


Figure 17 Armadillo noise synthesis and improvement results. Left: original, center: noise synthesis model, right: improved model using LDM.

Table 2 presents the quantitative results from the experiments shown in Figures 16 and 17. The Stanford Rabbit model with synthesized noise in the experiment showed an average error of 0.091 mm and a standard deviation of 1.084 mm. After applying the improvement model, the average error was reduced to 0.0415 mm, with a standard deviation of 0.37 mm. For the Armadillo model, the noisy version had an average error of -0.345 mm and a standard deviation of 0.454 mm. The improved model showed an average error of 0.005 mm and a standard deviation of 0.24 mm. These results

Table 2 Accuracy comparison of improved models using CloudCompare

	Mean			Standard Deviation		
Ground truth	Noise model	LDM	Ratio	Noise model	LDM	Ratio
Rabbit	0.091 mm	0.041 mm	54.8%	1.084 mm	0.37 mm	65.9%
Armadillo	−0.345 mm	0.005 mm	98.2%	0.454 mm	0.24 mm	47%

demonstrate that LDM effectively reduces noise and improves the quality of the mesh surface.

5 Conclusion

This paper proposes a calibration method for a distributed camera system using virtual viewpoints and a method for obtaining accurate camera extrinsic parameters for adaptive online transmission. The key feature of the proposed algorithm is its ability to randomly select frames from multiple viewpoints and iteratively minimize errors in feature point coordinates. The concept of a virtual viewpoint was employed in this process. Quantitative accuracy was measured through experiments conducted on two objects with known ground truth. The 3D Charuco board experiment reduced calibration errors to approximately 0.00926 mm. Considering physical distances, the proposed calibration method can almost precisely determine the relative positions between cameras. For the Moai statue, the accuracy of the alignment results was measured, and the experiment confirmed that the average error and standard deviation could be reduced to around 8 mm and 3.9 mm, respectively. Bilateral filtering and a point cloud refinement algorithm further enhanced the alignment results. Following this, mesh surface improvement using LDM was performed. By conditioning depth images obtained from the graphics pipeline and applying them to the inference process, experimental data showed that the average error was improved by 54.8% and the standard deviation by 65.9%. The proposed algorithm will be an excellent technique to control mesh quality for Web 3.0 graphic services such as WebGPU.

Acknowledgement

This research was supported by Seoul R & BD Program (CC240065) through the Seoul Business Agency (SBA) funded by Seoul Metropolitan Government. This research was supported by the MSIT (Ministry of Science and

ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2022-2020-0-01846) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

References

[1] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, 30(1):177–187, 2013.

[2] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.

[3] Matteo Munaro and Emanuele Menegatti. Fast rgb-d people tracking for service robots. *Autonomous Robots*, 37:227–242, 2014.

[4] Changhyun Choi and Henrik I Christensen. Rgb-d object tracking: A particle filter approach on gpu. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1091. IEEE, 2013.

[5] Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A textured object recognition pipeline for color and depth image data. In *2012 IEEE International Conference on Robotics and Automation*, pages 3467–3474. IEEE, 2012.

[6] Tewodros Legesse Munea, Yalew Zelalem Jembre, Halefom Tekle Weldegebriel, Longbiao Chen, Chenxi Huang, and Chenhui Yang. The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation. *IEEE Access*, 8:133330–133348, 2020.

[7] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652. Wiley Online Library, 2018.

[8] Silvio Giancola, Matteo Valenti, Remo Sala, Silvio Giancola, Matteo Valenti, and Remo Sala. State-of-the-art devices comparison. *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*, pages 29–39, 2018.

[9] Gozde Unal, Anthony Yezzi, Stefano Soatto, and Greg Slabaugh. A variational approach to problems in calibration of multiple cameras.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(8):1322–1338, 2007.

[10] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *sensors*, 12(2):1437–1454, 2012.

[11] Ilya V Mikhelson, Philip G Lee, Alan V Sahakian, Ying Wu, and Aggelos K Katsaggelos. Automatic, fast, online calibration between depth and color cameras. *Journal of Visual Communication and Image Representation*, 25(1):218–226, 2014.

[12] Aaron N Staranowicz, Garrett R Brown, Fabio Morbidi, and Gian-Luca Mariottini. Practical and accurate calibration of rgb-d cameras using spheres. *Computer Vision and Image Understanding*, 137:102–114, 2015.

[13] Kuisong Zheng, Yingfeng Chen, Feng Wu, and Xiaoping Chen. A general batch-calibration framework of service robots. In *Intelligent Robotics and Applications: 10th International Conference, ICIRA 2017, Wuhan, China, August 16–18, 2017, Proceedings, Part III 10*, pages 275–286. Springer, 2017.

[14] Jiyoung Jung, Joon-Young Lee, Yekeun Jeong, and In So Kweon. Time-of-flight sensor calibration for a color and depth camera pair. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1501–1513, 2014.

[15] Alina Kuznetsova and Bodo Rosenhahn. On calibration of a low-cost time-of-flight camera. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, pages 415–427. Springer, 2015.

[16] David Ferstl, Christian Reinbacher, Gernot Riegler, Matthias Rüther, and Horst Bischof. Learning depth calibration of time-of-flight cameras. In *BMVC*, pages 102–1, 2015.

[17] Alejandro Perez-Yus, Eduardo Fernandez-Moral, Gonzalo Lopez-Nicolas, Jose J Guerrero, and Patrick Rives. Extrinsic calibration of multiple rgb-d cameras from line observations. *IEEE Robotics and Automation Letters*, 3(1):273–280, 2017.

[18] Norishige Fukushima. Icp with depth compensation for calibration of multiple tof sensors. In *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2018.

[19] Byung-Seo Park, Woosuk Kim, Jin-Kyum Kim, Eui Seok Hwang, Dong-Wook Kim, and Young-Ho Seo. 3d static point cloud registration

by estimating temporal human pose at multiview. *Sensors*, 22(3):1097, 2022.

[20] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3945–3950. IEEE, 2007.

[21] Davide Scaramuzza, Ahad Harati, and Roland Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4169. IEEE, 2007.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[25] Microsoft. *Azure Kinect DK documentation*, 2021. Available at https://docs.microsoft.com/en-us/azure/kinect-dk/ (Accessed: 2021/02/03).

[26] Danil Kirsanov. *Minimal discrete curves and surfaces*. Harvard University, 2004.

[27] Xiang Ying, Xiaoning Wang, and Ying He. Saddle vertex graph (svg) a novel solution to the discrete geodesic problem. *ACM Transactions on Graphics (TOG)*, 32(6):1–12, 2013.

[28] Keenan Crane, Fernando De Goes, Mathieu Desbrun, and Peter Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, pages 1–126. 2013.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[30] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. Recent advances in remeshing of surfaces. *Shape analysis and structuring*, pages 53–82, 2008.

[31] Pierre Alliez, Eric Colin De Verdire, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *2003 Shape Modeling International.*, pages 49–58. IEEE, 2003.

[32] Jonathan Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, 2002, 2002.

[33] Yiqun Wang, Dong-Ming Yan, Xiaohan Liu, Chengcheng Tang, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. Isotropic surface remeshing without large and small angles. *IEEE transactions on visualization and computer graphics*, 25(7):2430–2442, 2018.

[34] Simone Melzi, Riccardo Marin, Pietro Musoni, Filippo Bardon, Marco Tarini, and Umberto Castellani. Intrinsic/extrinsic embedding for functional remeshing of 3d shapes. *Computers & Graphics*, 88:1–12, 2020.

[35] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.

[36] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (ToG)*, 26(3):96–es, 2007.

[37] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

Biographies

Byung-Seo Park received his B.A. degree in 2021 from the Dept. of Business Administration of Kwangwoon University in Seoul, Korea. He is a candidate for the Ph.D. program in Electronic Materials Engineering at Kwangwoon

University in Seoul, Korea. His research interests are 3D graphics, 2D and 3D image processing, and real time volumetric reconstruction.

Ye-Won Jang received her B.Sc. degree in Computer Engineering from Kwangwoon University in 2022 and is pursuing her M.Sc. degree in Electronic Materials Engineering at Kwangwoon University. Her research interests include 3D graphics, real-time motion capturing, and 3D model animating.

Hak-Bum Lee received his B.Sc. degree in Electronic Materials Engineering from Kwangwoon University in 2024 and is pursuing his M.Sc. degree in the same department. His research interests include multiview camera calibration for motion capture and 3D reconstruction of human motion.

Young-Ho Seo received his M.Sc. and Ph.D degrees in 2000 and 2004 from the Dept. of Electronic Materials Engineering of Kwangwoon University in Seoul, Korea. He was a researcher at the Korea Electrotechnology Research Institute (KERI) from 2003 to 2004 and was an assistant professor of Dept. of Information and Communication Engineering at Hansung University in Seoul, Korea. He has been a visiting professor at the University of Nebraska at Omaha, USA. He is now a full professor at the Department of Electronic Materials Engineering and a director of the Artificial Intelligence Research Center at Kwangwoon University in Seoul, Korea. His research interests include 3D graphics, 2D and 3D image processing, digital holography, real-time systems, and parallel processing.