

일반논문 (Regular Paper)

방송공학회논문지 제31권 제2호, 2026년 3월 (JBE Vol.31, No.2, March 2026)

<https://doi.org/10.5909/JBE.2026.31.2.260>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 대규모 고해상도 이미지의 3차원 공간영상 재구성을 위한 효율적인 파이프라인에 관한 연구

안 상 우<sup>a)</sup>, 임 한 신<sup>b)</sup>, 김 현 철<sup>a)</sup>, 권 혁 민<sup>a)</sup>, 추 현 곤<sup>a)†</sup>

### An Efficient Distributed Pipeline for Large-Scale High-Resolution 3D Spatial Image Reconstruction

Sangwoo Ahn<sup>a)</sup>, Hanshin Lim<sup>b)</sup>, Hyun-Cheol Kim<sup>a)</sup>, Hyukmin Kwon<sup>a)</sup>, and Hyon-Gon Choo<sup>a)†</sup>

#### 요 약

3차원 공간영상 서비스는 대규모 2차원 이미지로부터 고해상도의 사실적인 3D 환경을 재구성하는 서비스로서 메타버스, 문화유산 보존, 산업 모니터링 등 다양한 분야에 활용될 수 있다. 일반적인 3D 콘텐츠 생성 과정은 포인트 클라우드 생성, 메쉬 생성, 텍스처 매핑으로 구성되며 이와 관련한 연구가 활발히 진행되어 왔다. 그러나 기존 알고리즘과 처리 파이프라인은 처리 속도와 영상 품질의 한계가 존재한다. 이러한 문제를 해결하기 위해 본 논문은 대규모 고해상도 3D 공간 영상 재구성을 위한 효율적인 파이프라인을 제안한다. 제안된 시스템은 자동화된 이미지 그룹화와 분산 처리 방식을 지원하여 고밀도 포인트 클라우드 생성, 적응형 메쉬 재구성, 색상 연속성 기반 컬러 매핑을 수행한다. 이를 통해 대규모 데이터셋을 기존 방법보다 빠르게 처리하면서도 시각적 품질을 유지하거나 향상시킬 수 있다. 실험 결과, 제안된 파이프라인은 영상 품질과 처리 속도 모두에서 우수한 성능을 달성하였으며, 빠르고 정확한 3D 데이터 생성을 가능하게 한다. 본 연구는 다양한 실감형 서비스에서 효과적으로 활용될 수 있을 것으로 기대된다.

#### Abstract

Three-dimensional (3D) spatial image services reconstruct realistic and high-resolution 3D environments from large-scale collections of two-dimensional (2D) images, with applications spanning the metaverse, cultural heritage preservation, and industrial monitoring. The generation of 3D content typically involves point cloud construction, mesh generation, and texture mapping, and has been the subject of extensive research. However, existing algorithms and processing pipelines often suffer from limited processing speed and image quality. To overcome these limitations, this paper presents an efficient pipeline for large-scale and high-resolution 3D spatial image reconstruction. The proposed system supports automated image grouping and distributed processing for dense point cloud generation, adaptive mesh reconstruction, and color mapping based on color continuity. This design enables faster processing of massive datasets while preserving or enhancing visual quality compared to conventional methods. Experimental results demonstrate that the proposed pipeline achieves superior image quality while significantly improving processing speed. This approach facilitates fast and accurate 3D data generation and is expected to be effectively utilized in various immersive services.

Keyword : Point cloud, mesh, Color mapping, Spatial image generation, 3D spatial image service

Copyright © 2026 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

## I. Introduction

3D spatial imaging service uses 2D images acquired with still camera to reconstruct a more realistic 3D space and then provides it to users in various ways. The service makes it possible to reconstruct large areas of space, including precise restoration of small objects. It can accurately measure, record, and restore small objects as well as large spaces over time. For these reasons, demand for these more realistic 3D spatial image services has recently been rapidly increasing across various fields, including culture, industry, and games, including the metaverse [1, 2]. To meet this growing demand, related technology development is actively underway. The need to build 3D data for precise cultural heritage preservation and restoration further emphasizes the importance of these technologies [3]. The usability of cultural heritage can be greatly improved by precisely recording and preserving original cultural heritage data using point cloud-based 3D spatial imaging technology [4]. And, 3D spatial imaging services for wide-area spaces enable efficient data analysis in large-scale environments and can be used not only in industries such as construction and facility management, but also in more realistic metaverse services.

According to these demands, precise and efficient data processing technology in terms of optimized processing time for large-scale data and generation of high-quality 3D spatial images is essential. In [5], various surface reconstruction methods from point clouds were systematically categorized and compared, including Delaunay Triangulation, Poisson Reconstruction, and Implicit Function-based approaches. The paper evaluates their accuracy, computational efficiency, and robustness to noise, helping to determine the most suitable method for different applications. Similarly, [6] provides a comprehensive review of surface reconstruction techniques, focusing on Poisson-based reconstruction, Marching Cubes, and volumetric methods. It discusses the impact of point cloud density, noise levels, and sampling irregularities on reconstruction quality while highlighting the potential of deep learning-based approaches. In [7], an automated multi-view stereo (MVS) pipeline was developed to reconstruct 3D spaces from large-scale community image datasets. The system integrates point cloud generation, mesh construction, and texture mapping to create detailed 3D models. However, it struggles with diverse and unstructured image datasets, leading to lower resolution and incomplete surfaces, while also suffering from slow processing speeds due to the large-scale computations involved. In [8], a method for fast and accurate 3D reconstruction from multiple images was introduced. It enhances feature matching, depth estimation, and adaptive mesh refinement, improving both precision and computational efficiency. However, its performance is optimized for controlled environments and remains less effective in complex, real-world scenes with occlusions and varying lighting conditions. The above-mentioned studies are for different research directions and focus on presenting methodologies to solve specific problems. Especially, notably those represented by COLMAP [7][9][10][14], support a fully automated pipeline comprising feature extraction, matching, structure-from-motion (SfM), multi-view stereo (MVS), and

---

a) 한국전자통신연구원 실감미디어연구실(Immersive Media Research Laboratory, ETRI(Electronics and Telecommunications Research Institute))  
b) 동아대학교 컴퓨터공학부(Department of Computer Engineering, Dong-a University)  
‡ Corresponding Author : 추현곤(Hyon-Gon Choo)  
E-mail: hyongonchoo@etri.re.kr  
Tel: +82-42-860-1278  
ORCID: <https://orcid.org/0000-0002-0742-5429>

※ This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government with equal contributions (50% each) from the following projects: [26ZC1110 The research of the fundamental media contents technologies for hyper-realistic media space, RS-2025-02633263 Development of ultra-realistic low-latency video conferencing system technology based on light field display].  
· Manuscript January 12, 2026; Revised February 23, 2026; Accepted February 23, 2026.

mesh reconstruction, thereby enabling the generation of high-quality point clouds from input images. However, COLMAP suffers from limitations such as slow processing speed during the MVS stage and a lack of distributed processing support, which render it less suitable for applications involving large-scale 3D spatial image generation as proposed in this study. Recent neural reconstruction methods such as NeRF have achieved impressive photorealistic results, but they require heavy training data and computation, limiting scalability. In contrast, our pipeline emphasizes distributed efficiency and practical applicability for large-scale datasets.

Therefore, in order to satisfy the two requirements of optimized less processing time and high-quality 3D spatial image generation, this paper proposes the design and implementation of a point cloud-based 3D spatial image generation pipeline that distinctly stands out in comparison to previous approaches. The proposed pipeline integrates advanced algorithms for point cloud generation, adaptive meshing, and color mapping into a unified system that significantly enhances the efficiency and accuracy of 3D spatial reconstruction. By meticulously optimizing each of these algorithms, the pipeline not only achieves faster 3D reconstruction but also produces more refined and higher-quality 3D spatial images than conventional methods. One of the key differentiators of this approach is its incorporation of an automated distributed processing mechanism, which efficiently handles large-scale datasets while reducing computational bottlenecks. This mechanism enables a substantial reduction in processing time, particularly when dealing with massive, unstructured data, which is a known challenge in existing methods as shown in [7]. Furthermore, the pipeline implemented in this paper includes an innovative real-time monitoring feature, enabling users to evaluate intermediate outputs at each stage of processing. This feature allows for continuous iterative feedback, facilitating ongoing quality improvements throughout the workflow. In contrast to previous systems,

where quality control is often limited to final outputs or specific checkpoints, this real-time capability provides a more dynamic and adaptable approach to 3D reconstruction. The proposed 3D spatial imaging pipeline is capable of generating rapid, precise, and high-resolution 3D spatial images from 2D images captured by cameras, making it a significant advancement in the field. This capability enables the delivery of highly realistic and versatile 3D immersive imaging services, meeting the demands of a wide range of applications, including virtual reality, real-time modeling, and interactive spatial visualizations fields that require both speed and accuracy, and where many existing methods, as reviewed in [5] and [6], face limitations in terms of computational efficiency and quality consistency. In addition to improving reconstruction speed and quality, the proposed real-time monitoring feature is expected to enhance overall workflow efficiency by enabling early detection of errors and reducing unnecessary reprocessing. While this paper focuses on the design and implementation of the monitoring mechanism, a more detailed quantitative evaluation of its impact on actual work processes will be addressed in future research. The main contributions of this study are: (1) automatic image grouping with distributed SfM/MVS, (2) integration of adaptive meshing and vertex-based color mapping, and (3) incorporation of real-time monitoring to improve workflow efficiency.

## II. Proposed 3D reconstruction

### 1. System configuration

The point-cloud based 3D view generation process in this paper consists of point cloud generation function, adaptive meshing function, and color mapping function.

First, in the point cloud generation function, tens to thousands of images obtained from cameras are input and the

similarities of the images are analyzed. Images are grouped using the analyzed results, and a multi-pipeline is configured according to the number of groupings to enable distributed processing. Then, the feature points of each group image are extracted and a point cloud is generated by sequentially performing sparse reconstruction and dense reconstruction. Second, in the adaptive meshing function, mesh generation is performed by applying the geometric characteristics of point cloud data. In other words, a more efficient mesh is created by scanning the mesh plane and integrating similar planes into one. Third, in color mapping function, color mapping is performed according to an algorithm that can maintain the consistency of color values between adjacent vertexes.

Therefore, the point cloud-based 3D spatial image generation technology proposed in this paper integrates the following technologies to generate faster and more precise spatial images.

- Approach for configuring a multi-pipeline that automatically classifies and groups similar images from the acquired dataset, enabling parallel processing pipelines that improve the efficiency and speed of data handling for large-scale image collections.
- Point cloud generation approach that identifies and ex-

tracts key feature points from images, allowing for the generation of a dense point cloud with higher precision and resolution than conventional methods. This technique enhances the quality and accuracy of the 3D model by ensuring more accurate depth and spatial information.

- Adaptive meshing approach that analyzes the geometric properties of the dense point cloud to create a more efficient and optimized mesh. This process adjusts the mesh topology based on the underlying structure of the point cloud, ensuring smoother and more accurate surface representations.
- Color mapping approach that ensures visual consistency by performing a detailed analysis of the relationship between the generated mesh and the original images. This mapping ensures that texture and color are applied accurately across the model, preserving realistic visual continuity in the final 3D rendering.
- Distributed processing approach designed to reduce overall processing time by splitting the computational load across multiple servers. Each server handles a specific part of the multi-pipeline process, ensuring faster execution of the entire workflow while maintaining scalability for large datasets.

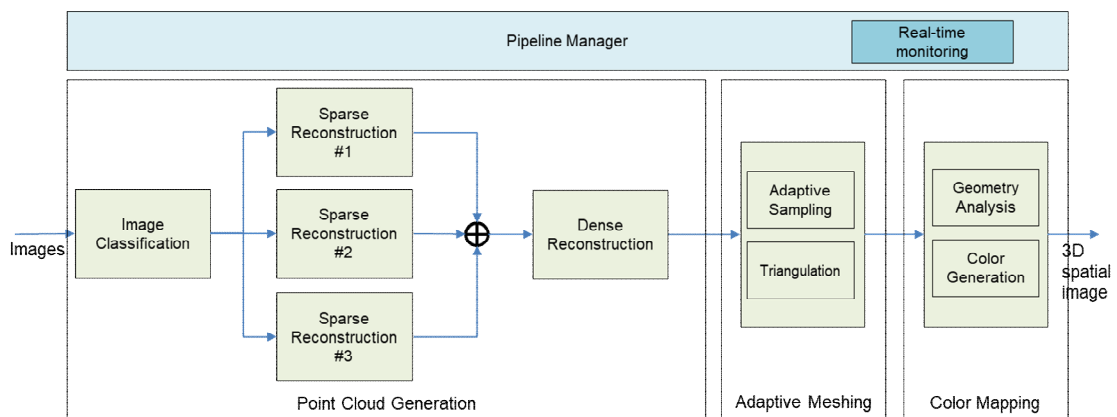


그림 1. 3D 공간영상을 생성하기 위하여 본 논문에서 제안한 기능 블록도  
 Fig. 1. The proposed block diagram of 3D spatial image generation

As mentioned above, the proposed pipeline consists of a pipeline manager, a point cloud generation module, an adaptive meshing module, and a color mapping module as described in Figure 1. To enable the generation of 3D spatial video content, we designed a system that allows users to construct various types of processing pipelines through an intuitive user interface. For efficient handling of large-scale image inputs, a distributed processing framework was implemented to rapidly generate point cloud data. To facilitate effective mesh generation from the resulting point clouds, we developed a plane-based meshing technique. Additionally, a color mapping method was designed to ensure continuity and coherence of color across the generated meshes. Furthermore, a real-time monitoring feature was integrated to allow continuous inspection of each processing stage.

To complement the block diagram in Figure 1 and to improve readability, Table 1 provides a step-by-step summary of the proposed pipeline and contrasts it with the COLMAP pipeline. This tabular representation highlights the structural differences between the two systems, allowing readers to easily grasp the unique contributions of our method.

The following section 2.2 provides a detailed explanation of point cloud generation, section 2.3 delves into adaptive meshing, and section 2.4 discusses color mapping in detail.

## 2. Point cloud generation

### 2.1 Sparse point cloud generation using distributed processing

As the number of image data increases, inefficiencies in generating sparse spatial data and quality degradation due to cumulative errors become significantly pronounced, highlighting the need for distributed processing through sub-grouping. For efficient processing of large-scale image groups, it is essential to rapidly sub-group the data, efficiently distribute and merge each sub-group, and manage the progress of each sub-group. To address this, a technology was developed to quickly group large-scale image data, generate sparse spatial data for each group in multiple servers, and merge the results. Figure 2 is the basic concept of the proposed distributed processing-based 3D reconstruction. Compared to COLMAP [9], the proposed method can generate the scene graph and sparse spatial data in a distributed manner.

The proposed method accelerates the scene graph generation process by utilizing multi-machine distributed processing during both the SIFT feature extraction and matching stages. In the feature extraction stage, the master PC partitions the input images according to the number of available GPUs and assigns them to multiple servers. Each server independently extracts SIFT feature points and generates a local database, which is later merged by the master

표 1. 본 논문에서 제시한 파이프라인과 COLMAP의 주요 기능 비교  
Table 1. Comparison of Proposed Pipeline and COLMAP

Step	Proposed Pipeline	COLMAP Pipeline
Input Data Handling	Automatic similarity analysis and grouping of images → multi-pipeline distributed processing	Single pipeline, limited scalability for large-scale datasets
Point Cloud Generation	Sparse + Dense reconstruction with distributed processing support	Sparse + Dense reconstruction in a single pipeline
Meshing	Adaptive meshing (plane-based optimization)	General mesh reconstruction without adaptive optimization
Color Mapping	Consistency-preserving algorithm ensuring coherent color values across adjacent vertices	Basic texture mapping applied after reconstruction
Monitoring	Real-time monitoring of intermediate outputs for iterative quality control	Quality control mainly limited to final outputs or specific checkpoints

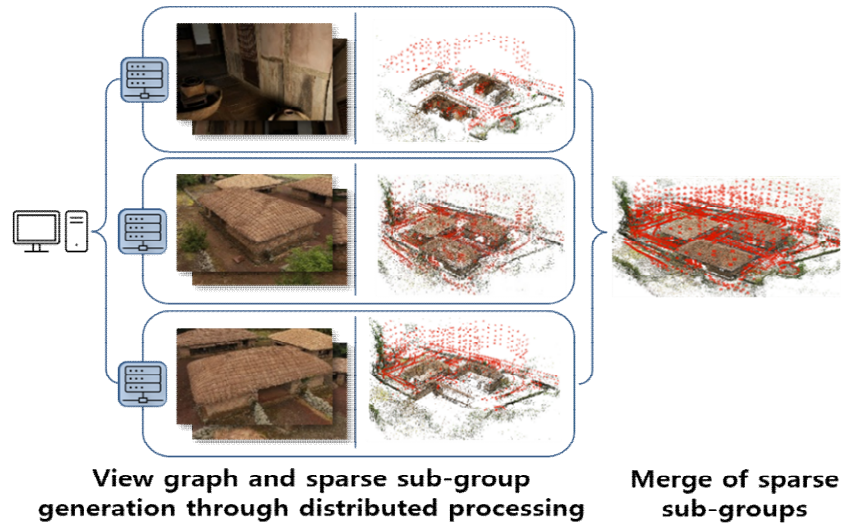


그림 2. 분산 처리 기반 3D 재구성의 기본 개념  
 Fig. 2. Basic concept of distributed processing-based 3D reconstruction

PC into a unified feature point database.

During the distributed matching stage, the master PC distributes the consolidated database to all servers, where exhaustive feature matching is performed between image pairs. To accelerate this computationally intensive step, the matching array is partitioned based on the number of GPUs, and each GPU conducts exhaustive matching on its assigned subset in parallel.

Following scene graph generation, the method also enhances the efficiency of sparse spatial data reconstruction by employing distributed processing combined with scene graph partitioning. The image set is divided into several groups, each assigned to one of NNN servers for independent sparse reconstruction. Scene graph partitioning is carried out on the master PC using the Normalized Cut algorithm as proposed in GraphSfM [10].

Once the sparse spatial data generation is complete, each group is transformed into a world coordinate system. Each subgroup's sparse spatial data exists in its own local coordinate system; therefore, to merge them, the datasets must be transformed into a common world coordinate system. To achieve this, a similarity transformation matrix

(comprising rotation, translation, and scale) must be estimated to convert the 3D feature points and camera poses of each sparse spatial dataset into the world coordinates.

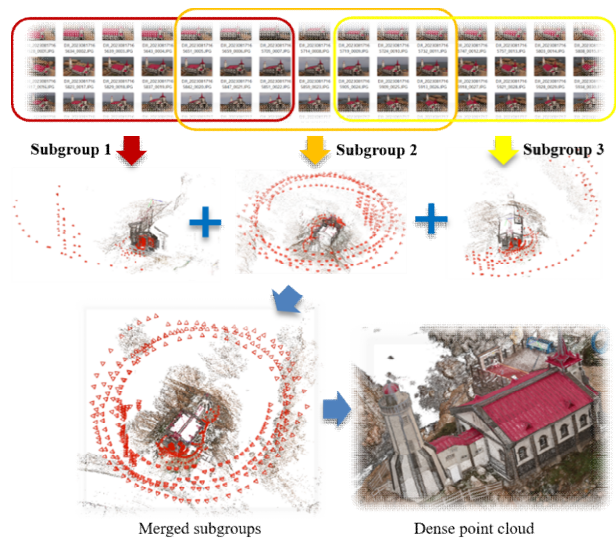


그림 3. 제한한 분산 처리 방법 및 조밀 공간 데이터 생성 방법을 사용, 축소 상당 데이터의 희소 공간데이터 재구성 및 조밀 공간데이터 생성 예  
 Fig. 3. An example of the sparse reconstruction and dense point cloud generation from the Jukseong church in Busan dataset by using the proposed distributed processing and the dense point cloud generation methods

In the merging process, the similarity transformation matrix is calculated using the gDLS (generalized Direct Least Squares) algorithm [11] with RANSAC based on the camera parameters and the 3D coordinates of features between the generated groups. After that, the obtained similarity transformation matrix is used to transform the camera parameters and the 3D coordinates of features of each group into the same coordinate system. During the merging of sparse spatial data, the merging order begins with subgroups that yield lower errors in similarity transformation.

Figure 3 shows an example of the sparse reconstruction and dense point cloud generation from the Jukseong church in Busan dataset by using the proposed distributed processing and the dense point cloud generation methods.

## 2.2 Dense point cloud generation based on multi-view stereo

After obtaining the sparse spatial data, a dense point cloud is generated for each group. Generating a dense point cloud seeks to accurately capture the structure of scenes and objects in multi-view images as a 3D point cloud by leveraging both the images and camera parameters. The most crucial step in this process is deriving high-quality dense depth maps from these images.

From the camera parameters computed in the sparse spatial data generation step, high-quality dense depth maps are obtained by combining depth maps generated using the traditional PatchMatch [12] technique with a triangular model.

PatchMatch approaches can achieve relatively accurate depth values in textured or detailed areas, while triangular

models more reliably represent surfaces in low-textured regions. ACMP [13] integrated a planar prior information to the matching cost in the PatchMatch process to utilize the strength of both approaches. However, because the solution space, or set of hypotheses, is not adjusted during the matching process, the planar prior is not fully integrated into the depth map generated by PatchMatch.

The proposed method reduces the solution space for each pixel, effectively simplifying the integration of the planar prior with the initial depth values computed by PatchMatch. By incorporating a sufficient planar prior into the depth map, the result serves as an improved initialization for subsequent PatchMatch processes [14].

Additionally, the proposed method refines the planar priors before the combination process, improving the reliability of the planar parameters [14].

Figure 4 shows the overall procedure of the proposed dense depth map generation method. In the depth map generation, first, initial depth and normal maps are obtained from the original images by the PatchMatch method. After that, a triangular model is generated from the selected nodes, and an initial planar prior is computed from the triangular model. The initial planar prior is then refined by applying a bilateral approach. The planar prior is combined with the initial depth and normal maps using a simple selection method [14]. Throughout the simple and efficient combining process, we can enhance depth estimation accuracy in low-textured areas, where existing PatchMatch-based methods have struggled, and in detailed regions, where the triangular model has limitations in surface representation.

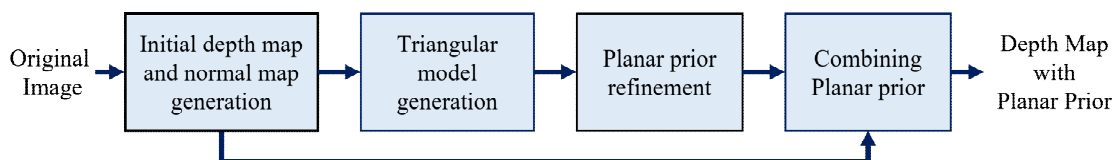


그림 4. 조밀 깊이 맵 생성 방법에 대한 전체 절차 [14]

Fig. 4. Overall procedure of the proposed dense depth map generation method [14]

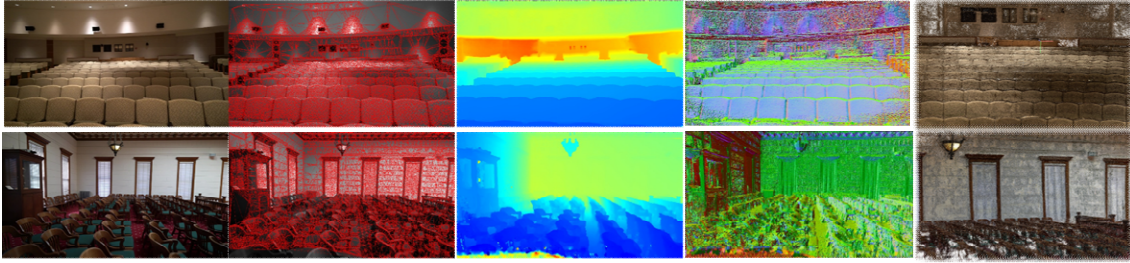


그림 5. 제안된 방법에 의해 생성된 컬러 이미지, 2D 메시, 깊이 맵, 노멀 맵, 포인트 클라우드 예  
 Fig. 5. Color images, 2D meshes, depth maps, normal maps, and point clouds generated by the proposed method

Figure 5 illustrates the color images, 2D meshes, depth maps, normal maps, and point clouds generated by the proposed method to the Tanks and Temples Advanced test set [15].

An accurate and dense point cloud can be generated by fusing the high-quality dense depth maps in the world coordinate.

### 3. Adaptive meshing generation using adaptive point cloud sampling

To generate a mesh from a point cloud, it is crucial to balance computational efficiency with the preservation of essential geometric details. While reducing the number of points improves efficiency, it is equally important to retain structural features that contribute to the accuracy and quality of the reconstructed mesh. Therefore, an effective sampling strategy should adaptively allocate point density based on local geometric complexity. We propose an adaptive sampling method that selectively reduces the point count while preserving critical geometric details necessary for precise reconstruction.

Traditional uniform down-sampling methods indiscriminately reduce point density, often leading to the loss of fine details in geometrically complex regions. Conversely, adaptive methods that lack a well-defined complexity measure may either retain excessive points in simple regions or oversimplify detailed structures. Our ap-

proach addresses these limitations by introducing a structured, complexity-aware sampling strategy based on an eigenvalue-driven geometric complexity metric.

To evaluate geometric complexity, we employ Principal Component Analysis (PCA) on the 3D points within each voxel, extracting the eigenvalues  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  (where  $\lambda_1 < \lambda_2 < \lambda_3$ ). We define complexity as

$$\text{Complexity} = \lambda_3 / (\lambda_2 - \lambda_1) \quad (1)$$

This metric quantifies the geometric variation of a local region. When  $\lambda_2$  and  $\lambda_1$  are similar, the local structure is nearly planar; a large difference between  $\lambda_3$  and  $(\lambda_2 - \lambda_1)$  indicates significant deviation from planarity, suggesting curved or irregular surfaces. While PCA-based eigenvalue ratios have been widely used to characterize local surface properties, our formulation focuses on the relative dispersion between the second and third principal directions, providing a more sensitive indicator of non-planarity in 3D point distributions.

We employ an octree-based voxelization method to adaptively partition the point cloud according to this complexity measure. The process begins by initializing the point cloud within a single encompassing voxel that contains all input points. Starting from this initial voxel, we recursively subdivide each voxel into eight smaller voxels if its complexity exceeds a predefined threshold  $\tau$ . This adaptive subdivision results in finer voxelization in

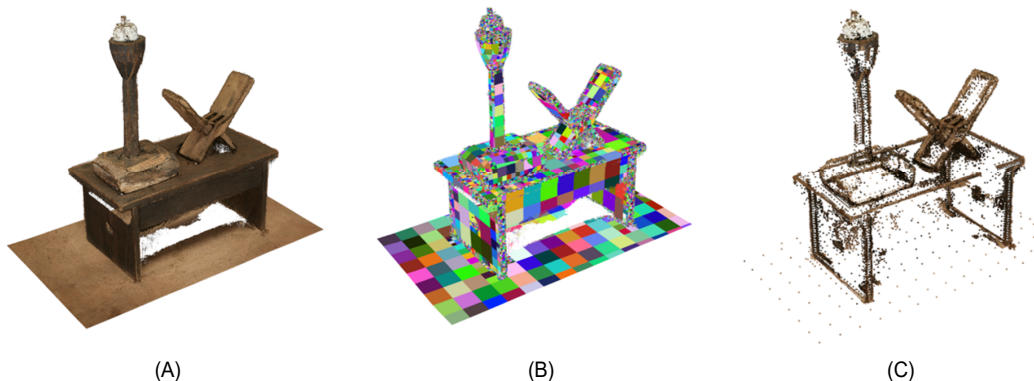


그림 6. (A) 입력 포인트 클라우드 (B) 옥트리 기반 복셀화 결과 (C) 포인트의 적응적 샘플링 결과  
 Fig. 6. (A) Input point cloud (B) Result of octree-based voxelization (C) Result of adaptive sampling of the point

high-complexity regions while preserving larger voxels in simpler, planar areas. Unlike fixed-grid voxelization, which enforces uniform spatial partitioning, our octree-based approach dynamically adjusts voxel sizes based on local geometric properties, ensuring a more meaningful spatial representation.

After the adaptive subdivision process converges, sampling is performed at the voxel level. Specifically, we select one representative point per voxel, typically the centroid of the points contained within it. This guarantees uniform coverage within each voxel while naturally preserving higher point density in geometrically complex regions due to finer subdivision. The resulting sampled point cloud thus retains essential structural details required for accurate mesh reconstruction.

We applied our octree-based voxelization method to sample an input point cloud before generating a mesh using the method described in [18]. The input point cloud, shown in Figure 6 (A), consists of 519,138 points. Figure 6 (B) illustrates the results of our proposed octree-based voxelization. After applying our adaptive sampling approach, which keeps a single representative point per voxel, the resulting point cloud, illustrated in Figure 6 (C), contains only 11,709 points—approximately 2.2% of the input point cloud. Despite this significant reduction, the resulting mesh (Figure 7 (B)) faithfully represents the actual object



그림 7. (A) 실제 객체의 사진, (B) 제안 방법을 사용해 생성한 3차원 모델  
 Fig. 7. (A) Photograph of the actual object (B) 3D model generated after sampling the point cloud using the proposed

(Figure 7 (A)), demonstrating that our approach preserves essential structural details while optimizing computational efficiency.

By integrating our complexity measure and octree-based adaptive sampling into the mesh generation pipeline, our method ensures a more efficient yet high-fidelity representation of 3D objects. This adaptive strategy overcomes the limitations of uniform down-sampling by effectively preserving essential structural features required for accurate reconstruction.

#### 4. Color mapping

Coloring a reconstructed 3D model is an important process to make the 3D model more realistic. However, automatically assigning colors from real-world RGB images

captured at various locations is challenging. Variations in color caused by differences in shooting locations or camera types can result in visible seams on the 3D model. Therefore, it is necessary to minimize seams that are boundaries with distinct differences in color caused by the assignment of different images to two neighboring faces by maintaining the consistency of assigned colors. In this chapter, we describe the color mapping process and the proposed vertex pixel value-based tone mapping method, which maintains color consistency across adjacent faces by enforcing vertex-level coherence.

Color mapping process consists of four main steps. First, the image regions that are mapped to each face, referred to as patches, are identified using camera parameters. Second, the visible patches are selected. These include patches corresponding to faces that are not occluded by other faces from the camera's viewpoint. To determine visibility, a depth buffer with the same resolution as the image plane is generated, and each face is projected onto it. During this process, only the faces with the smallest depth values at each pixel are retained, ensuring that occluded faces are excluded from the final visible set. Third, tone mapping is applied to ensure color consistency across patches associated with the same face. Finally, the face color is assigned by selecting the patch with the largest area among the tone-mapped image patches with modified colors. Since the tone mapped images generated in the previous step are used, the images are selected based on the patch area, without considering the color.

We propose a tone mapping method based on vertex pixel values to reduce seams at face boundaries by making the patches corresponding to the same face have similar colors. The proposed method ensures consistency in the pixel values of corresponding vertices across a set of patches that map to the same face, and then adjusts the internal pixel values of each patch based on these modified vertex values.

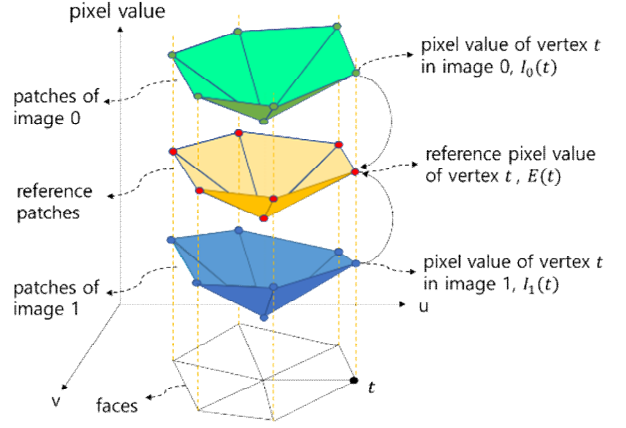


그림 8. 두 이미지의 정점 픽셀 값 기반 톤 매핑의 개념  
 Fig. 8. The concept of vertex pixel value-based tone mapping for two images

Figure 8 illustrates the concept of the proposed vertex pixel value-based tone mapping for two similar images of the same scene captured with different colors. In this example, 6 faces of a 3D model are depicted on the uv plane, and each image patch mapped to each face is represented by green and blue triangles, respectively. Here, the vertical axis represents the pixel value. The vertices of the yellow triangles are reference pixel values, which are the average of the pixel values of corresponding vertex in patch set, and can be expressed as Equation (2). The pixel value is applied to R, G, and B channels, respectively.

$$E(t) = \frac{1}{n} \sum_{i=1}^n I_i(t) \quad (2)$$

where,  $n$  is the number of images,  $I_i(t)$  is pixel value of vertex  $t$  projected to image  $i$ .

In order for a vertex to be a reference pixel value, the vertex pixel value needs to be modified by amount shown in Equation (3).

$$T_i(t) = E(t) - I_i(t) \quad (3)$$

This modification amount  $T_i(t)$  is used as the triangle

vertex value in the barycentric coordinate to correct the pixel value inside the triangle patch based on Equation (4). Equation (4) represents the process of adjusting the internal pixel values of each triangular patch based on the vertex modification amounts  $T_i(a)$ ,  $T_i(b)$ , and  $T_i(c)$ . Using barycentric interpolation, the modification amounts at the three vertices are linearly blended across the interior of the triangle. In this coordinate system, any point inside a triangle can be represented as a weighted combination of its three vertices, where the weights  $\alpha$ ,  $\beta$ , and  $\gamma$  represent the relative contribution of each vertex to the interpolated pixel value. As a result, pixels inside the patch are smoothly corrected so that their colors gradually transition according to the modified vertex values. This ensures color continuity across neighboring faces by aligning pixel intensities along shared vertices and edges, thereby reducing visible seams in the final 3D model.

$$O_i(x, y) = I_i(x, y) + \alpha T_i(a) + \beta T_i(b) + \gamma T_i(c) \quad (4)$$

where,  $\alpha + \beta + \gamma = 1$ ,  $(x, y)$  is the pixel coordinate inside the patch,  $I_i(x, y)$  is an input pixel value of  $(x, y)$ ,  $O_i(x, y)$  is a changed pixel value,  $T_i(a)$ ,  $T_i(b)$ ,  $T_i(c)$  represent the vertex pixel value modification amounts at vertices a, b, and c of the patch, respectively.

Figure 9 shows an example of images with colors modified by the proposed vertex pixel value-based tone mapping. In Figure 9, the image on the upper left was taken with a DSLR camera, while the image on the lower left was captured by a drone camera. Due to differences in their sensor types, there is a significant variation in color between the two images. The two images on the right with little color difference are the result of the proposed algorithm.

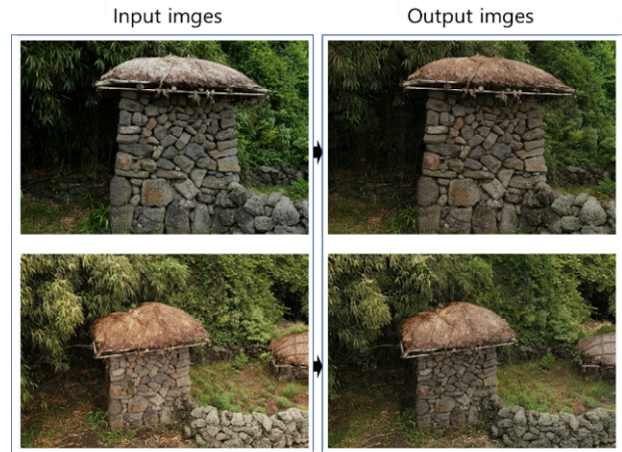


그림 9. 좌쪽 상단 및 하단은 촬영된 입력 이미지, 오른쪽 상단과 하단은 제안한 알고리즘을 사용하여 변환된 결과 이미지

Fig. 9. Top left and bottom left are input images taken with different cameras, top right and bottom right are images changed using proposed algorithm

### III. Implementation

The spatial image generation pipeline proposed in this paper was designed so that each functional block can be connected and configured according to the user interface. The parameters of each functional block (point cloud generation block, mesh generation block, and color mapping block) were set separately to enable flexible operation. In addition, the interface was configured to confirm normal operation by monitoring the result data of each block. This flexible configuration allows users to tailor the pipeline to specific requirements, optimizing the process based on the characteristics of the input data. Moreover, the real-time monitoring capability offers continuous feedback, ensuring that potential issues are detected early and adjustments can be made dynamically to maintain high accuracy and efficiency throughout the entire spatial image generation process. And the entire user interface screen consists of a menu area, execution area, pipeline area, and monitoring area, and the main functions of each are as follows.

- Menu area: Area for configuring a set of functional blocks such as Point cloud Generation block, Adaptive Meshing block, Color Mapping block, etc.
- Execution area: Area for executing the entire multi-pipeline or one pipeline that constitutes a multi-pipeline.
- Pipeline area: Area for configuring a pipeline by dragging blocks in the menu area with the mouse. In this domain, users are afforded the flexibility to construct various processing pipelines tailored to their specific requirements.
- Monitoring area: Area for outputting intermediate result images of each function block according to user selection.

Accordingly, the designed spatial image generation pipeline can be configured in various ways depending on the user interface, and the parameters required for each functional block can be set or changed through “Setting” of the corresponding block. A pipeline can be configured by porting commercial function blocks (tools) through an external interface, thereby directly comparing the performance of each tool. In addition, the final output spatial image was designed to support a variety of formats so that it can be played in a commercial rendering engine.

The above Figure 10 shows the result screen of the point cloud-based spatial image generation pipeline implemented according to the design described above. As shown in Figure 11 (A), the designed menu area, execution area, pipeline area, and monitoring area were implemented. In the menu area, there are various functional blocks including point cloud generation block, mesh generation block, color mapping block, etc. for creating 3D spatial images. And each functional block can port not only the algorithm proposed in this paper but also external commercial algorithms. In the pipeline area, a pipeline can be configured by dragging function blocks with the mouse. In the execution area, the configured pipeline can be executed, and in the monitoring area, the result data for each step can be check. Figure 10 (B) shows an example of parameter setting of a function block. Before the algorithms of each functional block are executed, related parameters must be set, and this was realized as shown in Figure 10 (A).

#### IV. Experiment

Experiments using the pipeline proposed in this paper were conducted on two aspects: process execution time and subjective image quality evaluation of the final generated



그림 10. (A) 제안한 파이프라인의 구현 결과 (B) 파라미터 세팅을 위한 사용자 인터페이스  
 Fig. 10. (A) Implementation result of the proposed pipeline (B) User interface for parameter setting

3D spatial image. In the first part of the experiment, we measured and compared the execution times of the point cloud generation and mesh generation blocks against those of COLMAP, using identical input content. For the color mapping block, no direct baseline was available; therefore, only the standalone processing time was recorded. Importantly, all measurements excluded file input/output overhead, focusing solely on the internal computational processing. The input dataset used for these experiments was obtained from the Jukseong Church site in Busan, selected for its complex architectural structure and photogrammetric relevance. In the second part of the experiment, a subjective image quality evaluation was conducted on spatial videos generated by the pipeline proposed in this paper. The Mean Opinion Score (MOS) was measured across a variety of spatial video samples, and the resulting MOS values were consistently assessed as indicating “good” visual quality.

1. Process execution time

1.1 Comparison of processing time

This section presents a comparative analysis of process-

ing times for three core modules of the proposed system: point cloud generation, mesh reconstruction, and color mapping. Furthermore, based on the individually measured results, we compared the total processing time of the proposed pipeline against baseline methods.

In point cloud generation, for the comparative experiment, a dataset of 1,078 images of Jukseong church in Busan was used. Because our distributed system consists of three servers with six GPUs, we set the number of groups N to three. Six RTX A6000 GPUs were used for the test. The number of images in each subgroup was set to 444, including overlapping images. The processing time from feature extraction to sparse reconstruction of COLMAP [7][9][10][14] was compared with the proposed distributed processing method, which utilizes distributed processing in the feature extraction, feature matching, and the sparse reconstruction stages. As shown in the results, the proposed method was approximately six times faster than the COLMAP default in terms of overall processing time. In particular, during the sparse reconstruction stage, the performance gap widened to about seven times due to

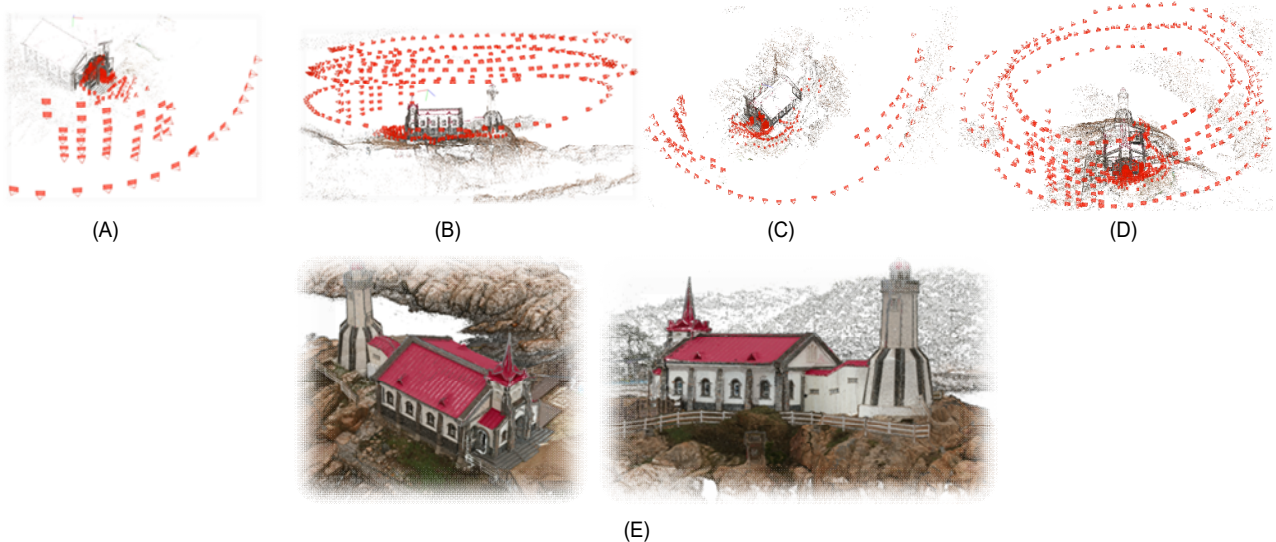


그림 11. (A), (B), (C)는 각 하위 그룹에 대해 생성된 희소 공간데이터 재구성 결과를, (D)는 병합된 희소 공간 데이터 재구성 결과를, (E)는 병합된 재구성을 입력으로 사용하여 생성된 최종 조밀 공간 데이터를 나타냄

Fig. 11. (A), (B), (C) Sparse reconstruction results generated for each subgroup, (D) the merged sparse reconstruction result, and (E) the resulting dense point cloud generated using the merged reconstruction as input

표 2. COLMAP과 제안한 방법에 대한 축소 성당 데이터의 특징 추출부터 희소 공간 데이터 재구성까지의 처리 시간 비교

Table 2. Comparison of the processing time from feature extraction to sparse reconstruction between COLMAP default and the proposed distributed processing method for Jukseong church in Busan dataset

		Feature extraction	Feature matching	Sparse reconstruction	Total
Processing time (min)	Colmap default	13.81	54.21	318.61	386.63
	Proposed distributed processing	6.75	14.90	44.47	66.12

표 3. 축소 성당 데이터의 COLMAP 메시 생성 [18]과 제안한 메시 생성 방법 간의 처리 시간 비교

Table 3. Processing Time Comparison Between COLMAP Mesh Generation [18] and Our Proposed Method

		Adaptive sampling	Triangulation	Simplification	Total
Processing time (min)	Colmap Mesh Generation	-	133.35	14.68	148.03
	Proposed Mesh Generation	1.06	38.01	2.16	41.23

the increased bundle adjustment time caused by the growing number of input images. After the sparse reconstruction, a 3D dense point cloud is generated using the proposed dense depth map reconstruction method described in Section 2.2.2. Figure 11 shows the sparse reconstruction results generated for each subgroup, the merged sparse reconstruction result, and the resulting dense point cloud generated using the merged reconstruction as input.

For mesh generation, we utilized the triangulation method provided in COLMAP [18]. We compared COLMAP mesh generation with our proposed adaptive sampling method described in Section 2.3.1. The number of faces in the final mesh output was fixed at 2 million for both methods. Both COLMAP and our method started with the same input point cloud containing approximately 33 million points. As shown in Table 2 and Table 3, COLMAP required 148.03 (min) to complete, whereas our proposed method only took 41.23 (min). This significant speedup is attributed to the reduction in the number of points prior to mesh generation. Importantly, as illustrated in Figure 12, the quality of the resulting mesh was either preserved or only slightly degraded. This is because our adaptive sampling effectively reduces points in a way that preserves the geometric complexity.

In color mapping, Table 4 shows the processing time of the proposed color mapping method using the Jukseong

표 4. 축소 성당 데이터의 제안한 컬러 매핑 처리 시간

Table 4. Processing time of the proposed color mapping for Jukseong church in Busan dataset

	Geometry Analysis	Color Generation	Total
Processing time of the proposed color mapping (sec)	516.5	261.4	777.9

church in Busan dataset. The processing time is broken down into “Geometry Analysis” and “Color Generation”. Geometry analysis, which takes 516.5 seconds, includes the first two steps of the color mapping process mentioned in Section 2.4: identifying image regions (patches) mapped to each face using camera parameters and selecting visible patches. Color generation, taking 261.4 seconds, covers the third and fourth steps: applying tone mapping for color consistency and assigning face colors by selecting the patch with the largest area among the tone-mapped images. The total processing time for the proposed color mapping is 777.9 seconds for a dataset consisting of 1078 images and 2 million faces. It's noted that COLMAP does not offer color mapping functionality, so no time measurements are provided for it.

As a result, the total processing time of the proposed pipeline was calculated by aggregating the execution times of each core module, as summarized in Table 5. As COLMAP lacks color mapping functionality, we performed color generation on the mesh generated by COLMAP using

표 5. COLMAP과 제안한 방법 간의 전체 처리 시간 비교  
 Table 5. Overall Processing Time Comparison Between COLMAP and Our Proposed Method

Processing time (min)	Point cloud generation		Mesh generation	Color mapping	Total
	Sparse	Dense			
COLMAP	318.61	384.56	148.03	12.93	864.13
Our Proposed Method	44.47	400.25	41.23	12.97	498.92

our proposed method. Compared to the baseline COLMAP system, the proposed pipeline achieved a processing time reduction of approximately 42.26%, demonstrating a significant improvement in computational efficiency.

1.2 Comparison of the qualitative results

This section presents a comparative analysis of the out-

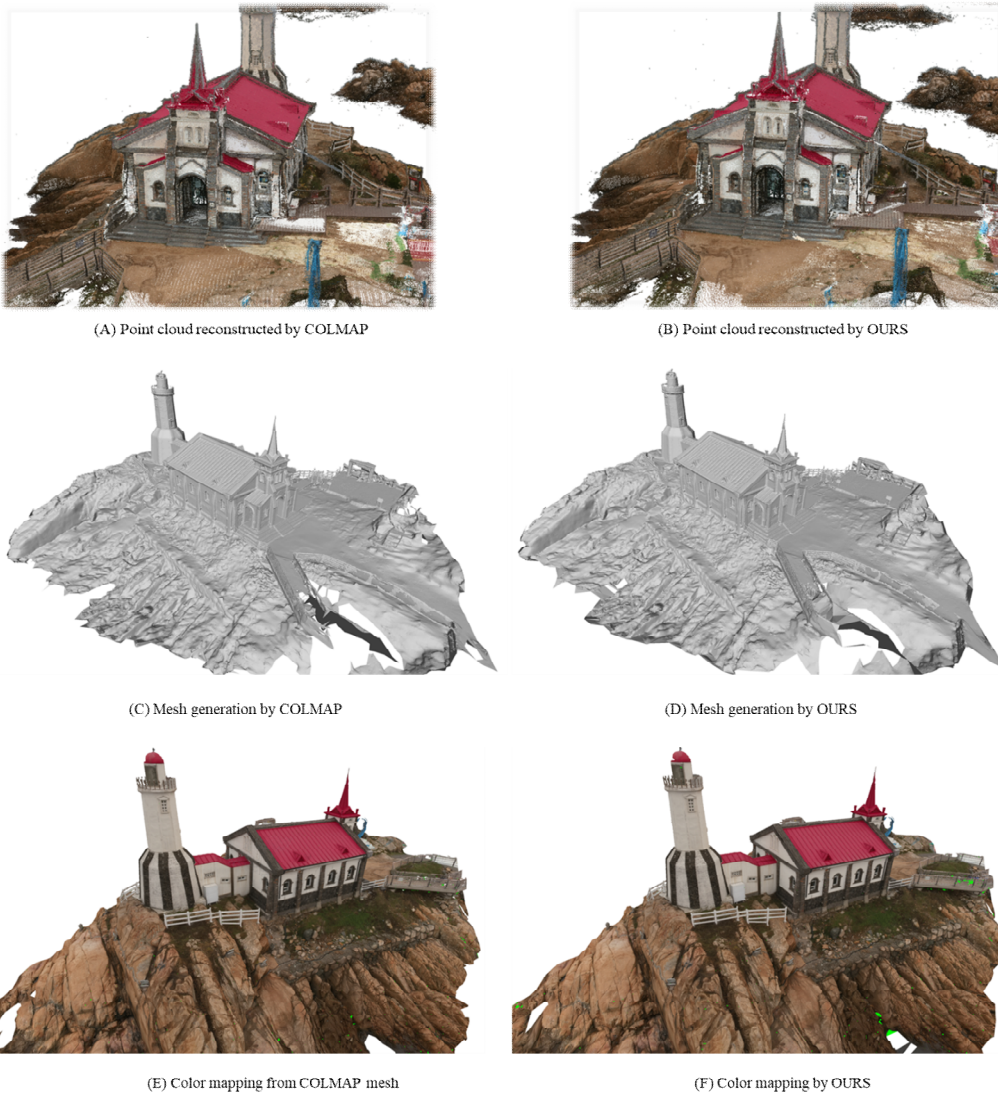


그림 12. 정성적 결과 비교: (A) COLMAP에 의해 재구성된 포인트 클라우드 (B) 제안한 방법에 의해 재구성된 포인트 클라우드 (C) COLMAP에 의한 메시 생성 (D) 제안한 방법에 의한 메시 생성 (E) COLMAP 메시로부터의 컬러 매핑 (F) 제안한 방법에 의한 컬러 매핑

Fig. 12. Comparison of the Qualitative Results: (A) Point cloud reconstructed by COLMAP (B) Point cloud reconstructed by ours (C) Mesh generation by COLMAP (D) Mesh generation by ours (E) Color mapping from COLMAP mesh (F) Color mapping by ours

put images generated by each core module of the proposed pipeline and those produced by COLMAP. Despite the notable reduction in processing time achieved by our method, subjective evaluations revealed no perceptible degradation in visual quality compared to COLMAP-generated results. Figure 12 illustrates the comparison: subfigures (A) and (B) show the point cloud outputs generated by COLMAP and our pipeline, respectively; (C) and (D) depict the mesh reconstruction results; and (E) and (F) present the color mapping outputs. It is important to note that COLMAP does not support color mapping in its standard pipeline; therefore, the result in subfigure (E) was exclusively obtained using the proposed method.

## 2. MOS quality assessment

This section presents the results of a subjective image

quality evaluation conducted on the output images generated by the proposed pipeline across a range of input content. For subjective image quality evaluation of the final generated 3D spatial image proposed in this paper, 15 contents on 3 themes were used as original images. Each original image was produced as half point cloud data, full point cloud data, and 3D spatial image. Accordingly, a subjective image quality evaluation was performed on a total of 45 contents. The following Figure 13 shows 3 themes and 15 original videos.

The subjective image quality evaluation algorithm was performed by measuring the MOS Mean Opinion Score of the original video and the comparison video based on SAMVIQ (Subjective Assessment Methodology for Video Quality), one of the subjective image quality evaluation methods described in ITU-R Recommendation BT.500, 1788 [20]. This method involves having participants rate



(A)



(B)



(C)

그림 13. 3개 테마영상에 대한 주관적 화질 평가: (A) 경주국립박물관 (B) 서대문형무소 (C) 드라마 세트장(죽성 성당)

Fig. 13. 3 themes for subjective image quality evaluation: (A) GYEONGJU National Museum (B) SEODAEMUN Prison History Hall (C) Drama set (JUKSEONG Church)

표 6. 3개 테마, 15개의 콘텐츠 영상에 대한 SAMVIQ, MOS 주관적 화질평가 결과. A는 1/2 포인트 클라우드 데이터, B는 Full 포인트 클라우드 데이터, C는 3D 공간영상 데이터를 나타냄

Table 6. Subjective image quality evaluation SAMVIQ and MOS results for 3-themes, 15 contents. A is half point cloud data, B is full point cloud data, and C is 3D spatial image from proposed algorithm

Categories		SAMVIQ	MOS
Theme #1	A	63.533	3.177
	B	74.600	3.730
	C	82.267	4.113
Theme #2	A	41.600	2.080
	B	49.867	2.493
	C	78.533	3.927
Theme #3	A	49.600	2.480
	B	53.000	2.650
	C	81.000	4.050
Average	A	51.578	2.579
	B	59.156	2.958
	C	80.600	4.030

the video quality based on their perceived experience of various visual elements, such as clarity, color consistency,

and overall sharpness. And, under the condition of a 75-inch TV and a 3-meter viewing distance, a subjective picture quality evaluation was performed on a group of 30 non-experts. The participants were carefully selected to ensure a broad range of opinions on the video quality. The following Table 6 shows the SAMVIQ and MOS results, which are the subjective image quality evaluation results [19, 20, 21].

The MOS evaluation was conducted with 30 non-expert participants to ensure diverse perspectives. The average MOS score for the proposed 3D spatial images was 4.03 with a standard deviation of 0.08, indicating consistent quality perception among participants. As shown in the Table 6, the evaluation scores of the produced comparative images, A. half point cloud and B. full point cloud, were calculated low, but the output 3D spatial image C of the pipeline proposed in this paper was calculated high with MOS 4.03 average. A value of MOS 4.0 or higher corre-

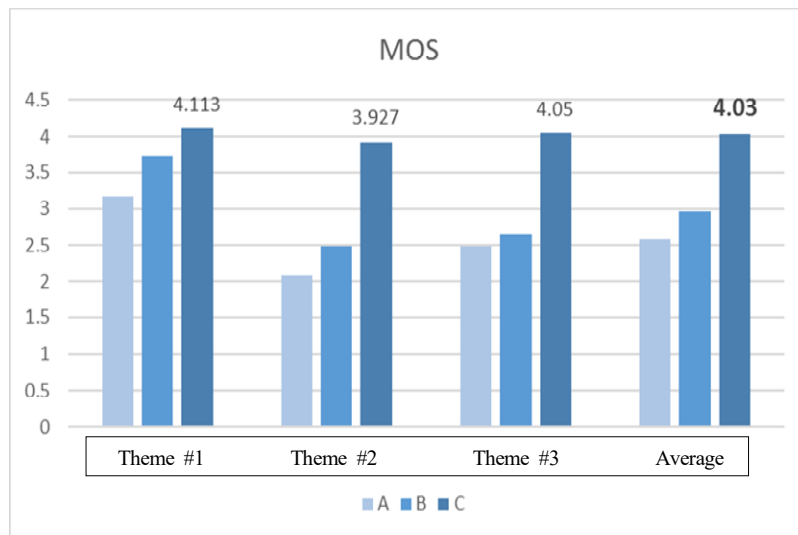


그림 14. 주관적 화질평가 결과: 각 그래프는 테마영상 1, 2, 3에 대한 MOS 측정 결과 값을 나타내며 평균 MOS 값도 제시함. 각 테마에 대해서, 첫 번째 값 A는 원본과 비교한 반해상도 포인트 클라우드의 MOS 점수를 나타내고, 두 번째 값 B는 원본과 비교한 전체 해상도 포인트 클라우드의 MOS 점수를 의미하며, 세 번째 값 C는 본 논문에서 제안한 메시 생성 및 색상 매핑 방법을 사용하여 얻은 결과의 MOS 점수를 나타냄

Fig. 14. Subjective image quality evaluation: The graph presents the measured MOS values for Theme #1, #2, and #3, along with the average MOS value. For each theme, the first value A represents the MOS score for the half-resolution point cloud compared to the original, the second value B corresponds to the MOS score for the full-resolution point cloud compared to the original, and the third value C indicates the MOS score for the results obtained using the proposed mesh generation and color mapping method in this paper

sponds to PSNR 35dB or higher, so it can be considered a high level of quality [19-24]. Therefore, it is expected that it can be used for photorealistic 3D video services in the near future. The following Figure 14 is a graph showing the SAMVIQ and MOS results, which are the subjective image quality evaluation results. It can be confirmed that the output 3D spatial image C of the pipeline proposed in this paper was calculated as high as MOS 4.03 average. The MOS score of 4.03 indicates that the proposed method achieves a photorealistic quality that is well-suited for high-fidelity 3D video services.

## V. Conclusion

In this study, we presented the design and implementation of a point cloud-based 3D spatial image generation pipeline to provide 3D realistic image services for wide-area spaces. The proposed pipeline can be composed of one or more pipelines by connecting functional blocks for generating 3D spatial images. Additionally, pipelines can be composed of multiple pipelines to support distributed processing. To achieve this, input images must be able to be classified based on similarity. Then the classified data must be distributed and processed through multiple pipelines. As mentioned above, in this paper, we designed and implemented algorithms for similarity classification, point cloud generation, adaptive meshing, and color mapping as well as algorithms for pipeline construction. We tested to measure speed and image quality throughout the entire process. To describe it in more detail, the experiment was performed by creating a pipeline using the implemented algorithm and user interface, and then focusing on evaluating the process time and image quality it took to generate a 3D spatial image using the acquired images as input. As a result of the experiment, we confirmed the normal operation of the pipeline without any errors. First, although the process time depends on the type of input con-

tent and the number of distributed processing according to the multi-pipeline configuration, it was confirmed that processing performance is about 1.6 times faster than that of commercial tools. Secondly, it was confirmed that the image quality of the generated 3D spatial image achieved an MOS index value of 4.03 as a subjective image quality evaluation.

The pipeline proposed in this paper can be used more usefully in generating 3D spatial images in a wide area because it can be configured by connecting various existing functional blocks into a single or multiple pipelines. Additionally, each functional block is developed with an optimized algorithm, which has the advantage of maintaining faster speed and higher quality. This pipeline can significantly reduce the processing time for cultural heritage preservation and metaverse applications, making high-precision 3D reconstruction more accessible [4]. The maintenance of high quality and convenience of spatial image generation by the method proposed in this paper can enable 3D spatial image creation to be performed smoothly, and are expected to quickly meet the demand for 3D spatial image services, including Metaverse. Although the proposed pipeline demonstrated improved processing speed and image quality compared to commercial tools, further statistical validation (e.g., significance testing) will be conducted in future work to strengthen the evidence of its superiority. In addition, brief clarifications have been provided regarding evaluator composition, variance in MOS results, and threshold selection in adaptive sampling, while more extensive comparisons with recent learning-based approaches will be addressed in future research.

## References

- [1] F. Gao, G.-W. Gao, D. Wu, and F. Wang, "Research on 3D periodontal reconstruction algorithm based on point cloud reconstruction," (Proc. 2022 4th Int. Conf. Intell. Control, Meas. Signal Process. (ICMSP), 2022), pp. 1032 - 1035.

- doi: <https://doi.org/10.1109/ICMSP55950.2022.9859147>
- [2] X. Zhang, J. Liu, B. Zhang, L. Sun, Y. Zhou, Y. Li, J. Zhang, H. Zhang, and X. Fan, "Research on object panoramic 3D point cloud reconstruction system based on structure from motion," *IEEE Access*, 10 (2022), pp. 1 - 15.  
doi: <https://doi.org/10.1109/ACCESS.2022.3213815>
- [3] C. Wang, Y. Gu, and X. Li, "A robust multispectral point cloud generation method based on 3-D reconstruction from multispectral images," *IEEE Trans. Geosci. Remote Sens.*, 61 (2023), pp. 1 - 12.  
doi: <https://doi.org/10.1109/TGRS.2023.3326153>
- [4] Y. Wang, P. Zhou, G. Geng, L. An, and M. Zhou, "Enhancing point cloud registration with transformer: Cultural heritage protection of the Terracotta Warriors," *Heritage Sci.*, 12 (2024), no. 1, pp. 25 - 35.  
doi: <https://doi.org/10.1186/s40494-024-01425-9>
- [5] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Comput. Graph. Forum*, 36 (2017), no. 1, pp. 301 - 329.  
doi: <https://doi.org/10.1111/cgf.12802>
- [6] F. Calakli and G. Taubin, "SSD: Smooth Signed Distance Surface Reconstruction," *Comput. Graph. Forum*, 37 (2018), pp. 1993 - 2002.  
doi: <https://doi.org/10.1111/j.1467-8659.2011.02058.x>
- [7] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys, "Multi-view stereo for community photo collections," (Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2016), pp. 1481 - 1490.  
doi: <https://doi.org/10.1109/ICCV.2007.4408933>
- [8] Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun, "Pix2Vox++: Multi-scale context-aware 3D object reconstruction from single and multiple images," *International Journal of Computer Vision*, Vol.128, No.12, pp.2919 - 2934, 2020.  
doi: <https://doi.org/10.1007/s11263-020-01347-6>
- [9] C. Sweeney, V. Fragoso, T. Hollerer, and M. Turk, "gDLS: A scalable solution to the generalized pose and scale problem," (Proc. Eur. Conf. Comput. Vis. (ECCV), 2014), pp. 16 - 31.  
doi: [https://doi.org/10.1007/978-3-319-10593-2\\_2](https://doi.org/10.1007/978-3-319-10593-2_2)
- [10] J. L. Schönberger and J. M. Frahm, "Structure-from-motion revisited," (Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), 2016), pp. 4104 - 4113.  
doi: <https://doi.org/10.1109/CVPR.2016.445>
- [11] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patch-Match: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, 28 (2009), no. 3, Article No. 24, pp. 1 - 11.  
doi: <https://doi.org/10.1145/1531326.1531330>
- [12] H. Lim and H. Choo, "A Simple and Efficient Method to Combine Depth Map with Planar Prior for Multi-View Stereo," *Journal of Broadcast Engineering*, 28(2023), pp.925-930.  
doi: <https://doi.org/10.5909/JBE.2023.28.7.925>
- [13] A. Knapitsch, J. Park, Q. Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, 36 (2017), no. 4, pp. 78:1 - 78:13.  
doi: <https://doi.org/10.1145/3072959.3073599>
- [14] J. L. Schönberger, E. Zhang, J. M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," (Proc. Eur. Conf. Comput. Vis. (ECCV), 2016), pp. 501 - 518.  
doi: [https://doi.org/10.1007/978-3-319-46487-9\\_31](https://doi.org/10.1007/978-3-319-46487-9_31)
- [15] Q. Xu and W. Tao, "Planar prior assisted PatchMatch multi-view stereo," (Proc. AAAI Conf. Artif. Intell., 2020), pp. 12516 - 12523.  
doi: <https://doi.org/10.1609/aaai.v34i07.6940>
- [16] F. Qin, Y. Ye, G. Huang, X. Chi, Y. He, X. Wang, Z. Zhu, and X. Wang, "MVSTER: Epipolar transformer for efficient multi-view stereo," (Proc. Eur. Conf. Comput. Vis. (ECCV), 2022), pp. 528 - 544.  
doi: <https://doi.org/10.48550/arXiv.2204.07346>
- [17] R. Peng, R. Wang, Z. Wang, Y. Lai, and R. Wang, "Rethinking depth estimation for multi-view stereo: A unified representation," (Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), 2022), pp. 8645 - 8654.  
doi: <https://doi.org/10.1109/CVPR52688.2022.00845>
- [18] P. Labatut, J.-P. Pons, and R. Keriven, "Robust and efficient surface reconstruction from range data," *Comput. Graph. Forum*, 28 (2009), no. 1, pp. 2275 - 2290.  
doi: <https://doi.org/10.1111/j.1467-8659.2009.01530.x>
- [19] Baek, M. Kim, S. Son, S. Ahn, J. Seo, H. Y. Kim, and H. Choi, "Visual quality assessment of point clouds compared to natural reference images," *J. Web Eng.*, 22 (2023), no. 3, pp. 405 - 432.  
doi: <https://doi.org/10.13052/jwe1540-9589.2232>
- [20] ITU-R BT.500-13, Methodology for the subjective assessment of the quality of television pictures, International Telecommunication Union Radiocommunication Sector, 2012. <https://www.itu.int/rec/R-REC-BT.500>
- [21] ITU-R BT.2021, Subjective methods for the assessment of stereoscopic 3DTV systems, International Telecommunication Union Radiocommunication Sector, 2012. <https://www.itu.int/rec/R-REC-BT.2021>
- [22] E. Dumic, F. Battisti, M. Carli, and L. A. da Silva Cruz, "Point cloud visualization methods: A study on subjective preferences," (Proc. Eur. Signal Process. Conf. (EUSIPCO), 2020), pp. 595 - 599.  
doi: <https://doi.org/10.23919/EUSIPCO47968.2020.9287504>
- [23] Y. Jin, M. Chen, T. Goodall, et al., "Subjective and objective quality assessment of 2D and 3D foveated video compression in virtual reality," *IEEE Trans. Image Process.*, 30 (2021), pp. 5905 - 5919.  
doi: <https://doi.org/10.1109/TIP.2021.3087322>
- [24] Q. Liu, H. Yuan, H. Su, et al., "PQA-Net: Deep no-reference point cloud quality assessment via multi-view projection," *IEEE Trans. Circuits Syst. Video Technol.*, 31 (2021), pp. 4645 - 4660.  
doi: <https://doi.org/10.1109/TCSVT.2021.3100282>

---

저 자 소 개

---



**안 상 우**

- 1999년 : 경희대학교 전자공학과 석사
- 1999년 ~ 현재 : 한국전자통신연구원 책임연구원
- ORCID : <https://orcid.org/0009-0007-0129-7685>
- 주관심분야 : 실감미디어, 컴퓨터 비전, 3D 영상처리, 라이트필드



**임 한 신**

- 2004년 2월 : 연세대학교 전기전자공학부 (수학 부전공) (공학사)
- 2006년 2월 : 한국과학기술원 전기전자공학부 (공학석사)
- 2007년 9월 ~ 2007년 12월 : TU Berlin 방문연구원
- 2014년 2월 : 한국과학기술원 전기전자공학부 (공학박사)
- 2014년 3월 ~ 2025년 8월 : 한국전자통신연구원 책임연구원
- 2025년 9월 ~ 현재 : 동아대학교 조교수
- ORCID : <https://orcid.org/0000-0003-4829-2893>
- 주관심분야 : Computer Vision, 3D Reconstruction and Modeling, VR/AR/XR Technology



**김 현 철**

- 2000년 8월 : 경희대학교 공학석사
- 2015년 2월 : 충남대학교 공학박사
- 2000년 7월 ~ 현재 : 한국전자통신연구원 초실감메타버스연구소 책임연구원
- ORCID : <https://orcid.org/0000-0001-6843-5949>
- 주관심분야 : 실감미디어, 컴퓨터비전, 3D 재구성



**권 혁 민**

- 2019년 : 고려대학교 전기전자공학과 석사
- 2019년 ~ 현재 : 한국전자통신연구원 선임연구원
- ORCID : <https://orcid.org/0000-0002-2161-1471>
- 주관심분야 : 실감미디어, 3차원 영상처리



**추 현 곤**

- 2005년 : 한양대학교 전자통신공학과 박사
- 2005년 ~ 현재 : 한국전자통신연구원 책임연구원
- 2024년 ~ 현재 : 한국전자통신연구원 실감미디어연구실장
- 2017년 ~ 2018년 : 바르샤바공과대학 방문연구원
- ORCID : <https://orcid.org/0000-0002-0742-5429>
- 주관심분야 : 실감미디어, 머신비전, 라이트필드, 홀로그램, 방송시스템