International Journal of
Geo-Information

Article

Performance Evaluation and Optimization of 3D Gaussian Splatting in Indoor Scene Generation and Rendering

Xinjian Fang *, Yingdan Zhang, Hao Tan, Chao Liu and Xu Yang

School of Spatial Information and Surveying and Mapping Engineering, Anhui University of Science and Technology, Huainan 232001, China; 2022201790@aust.edu.cn (Y.Z.); 2021217@aust.edu.cn (H.T.); chliu1@aust.edu.cn (C.L.); xyang@aust.edu.cn (X.Y.)

* Correspondence: fangxinjian@aust.edu.cn

Abstract: This study addresses the prevalent challenges of inefficiency and suboptimal quality in indoor 3D scene generation and rendering by proposing a parameter-tuning strategy for 3D Gaussian Splatting (3DGS). Through a systematic quantitative analysis of various performance indicators under differing resolution conditions, threshold settings for the average magnitude of spatial position gradients, and adjustments to the scaling learning rate, the optimal parameter configuration for the 3DGS model, specifically tailored for indoor modeling scenarios, is determined. Firstly, utilizing a self-collected dataset, a comprehensive comparison was conducted among COLLI-SION-MAPping (abbreviated as COLMAP (V3.7), an open-source software based on Structure from Motion and Multi-View Stereo (SFM-MVS)), Context Capture (V10.2) (abbreviated as CC, a software utilizing oblique photography algorithms), Neural Radiance Fields (NeRF), and the currently renowned 3DGS algorithm. The key dimensions of focus included the number of images, rendering time, and overall rendering effectiveness. Subsequently, based on this comparison, rigorous qualitative and quantitative evaluations are further conducted on the overall performance and detail processing capabilities of the 3DGS algorithm. Finally, to meet the specific requirements of indoor scene modeling and rendering, targeted parameter tuning is performed on the algorithm. The results demonstrate significant performance improvements in the optimized 3DGS algorithm: the PSNR metric increases by 4.3%, and the SSIM metric improves by 0.2%. The experimental results prove that the improved 3DGS algorithm exhibits superior expressive power and persuasiveness in indoor scene rendering.

Keywords: 3DGS; indoor scene rendering; performance evaluation; parameter optimization

check for updates

Academic Editors: Wolfgang Kainz, Junxing Zheng and Peng Cao

Received: 22 October 2024
Revised: 28 December 2024
Accepted: 2 January 2025
Published: 7 January 2025

Citation: Fang, X.; Zhang, Y.; Tan, H.; Liu, C.; Yang, X. Performance Evaluation and Optimization of 3D Gaussian Splatting in Indoor Scene Generation and Rendering. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 21. https://doi.org/10.3390/ijgi14010021

1. Introduction

Three-dimensional (3D) models of indoor environments are of paramount importance for a variety of applications such as navigation assistance and emergency response [1]. However, unlike their outdoor counterparts, the 3D reconstruction of indoor environments still poses specific challenges due to the nature of the complicated layout of the indoor structure, the complex interactions between objects, clutter, and occlusions [2]. In the context of indoor 3D scene generation and rendering, several primary algorithms are used. These include oblique photogrammetry, SFM-MVS [3], NeRF [4], and the noTable 3DGS [5] algorithm.

In the realm of traditional oblique photogrammetry algorithms, particularly those employing RGB-D camera technology, Yuan Zhilu et al. [6] conducted a comprehensive review of recent advancements and verified their potential through the analysis of various datasets. In the interim, Li Jikun and colleagues [7] conducted a study on the utilization of drone imagery in conjunction with Context Capture software (V10.2) for urban modeling and employed DP-Modeler to improve the quality of individual models. Jin Ye et al. [8] propose a 3D simulation system for indoor soft decoration based on OpenGL and Direct3D. Zavar H et al. [9] introduce a topology-aware data-driven approach for 3D reconstruction of indoor spaces. Although these studies primarily focus on the application and innovation of conventional algorithms, the issue of efficiency remains a critical challenge. Wang Jun et al. [10] proposed a novel framework aiming to reconstruct a high-quality, globally consistent 3D model for indoor environments using only an RGB-D sensor. An automatic indoor reconstruction method that quickly and effectively reconstructs the indoor environment of multi-floors and multi-rooms using both point clouds and trajectories from mobile laser scanning (MLS) is proposed [11].

In the field of Geographic Information Technology, BIM technology has become increasingly mature. The model construction and data organization of indoor maps are the key scientific problems that urgently need to be solved in the current indoor LBS application [12]. Poux, F et al. [13] propose a new shape grammar approach for the efficient generation of 3D models of indoor environments from point clouds. An overview of the geometric, semantic, and topological reconstruction of the indoor environment is presented, where the existing methodologies, advantages, and disadvantages of these three reconstruction types are analyzed and summarized [14].

In the domain of vision-based three-dimensional (3D) reconstruction, Structure from Motion (SFM) has gained considerable traction due to its capacity for automated camera tracking and motion estimation. Nevertheless, in practical applications, SFM-based 3D reconstruction methodologies are vulnerable to various influences, including image quality and the effectiveness of matching algorithms [15]. The SFM-MVS algorithm, recognized for its remarkable data processing efficiency, faces limitations when reconstructing typical scenes, reflecting challenges akin to those encountered with oblique photogrammetry techniques. In intricate scenarios characterized by reflections, low-texture regions, and occlusions, the SFM-MVS approach may encounter significant obstacles that lead to matching failures or suboptimal reconstruction outcomes. Dong Chen et al. [16] introduce a framework for reconstructing fine-grained room-level models from indoor point clouds.

The rapid advancements in neural rendering have led to the emergence of the concept of NeRF [17], seamlessly integrating deep learning into the domain of three-dimensional modeling and heralding a significant transformation in voxel reconstruction techniques. NeRF can render photorealistic 3D scenes [18]. Neural Radiance Fields (NeRFs) offer a state-of-the-art quality in synthesizing novel views of complex 3D scenes from a small subset of base images [19]. Jiakai Cao et al. [20] introduce NeRF-based Polarimetric Multi-view Stereo (NPMVS), a novel 3D reconstruction method that combines the advantages of neural radiance field (NeRF) and shape-from-polarization (SFM) to address the challenge posed by textureless areas while preserving the fine-scale geometric details. NeRF demonstrates exceptional capability in tackling intricate scenes featuring reflections, low-texture areas, and occlusions, ultimately enabling the creation of highly realistic virtual environments that enhance the vividness, authenticity, and seamless flow of visual presentations. Nevertheless, the challenge of prolonged training times persists as a notable challenge. In a groundbreaking move in 2021, Shihao Qin et al. [21] method incorporates the advantages of Zip-NeRF and incorporates depth information to reduce the number of required images and solve the scale-free problem in borderless scenes. Thomas et al. [22] introduced hash encoding into NeRF, significantly curtailing the training duration. Despite this commendable enhancement, NeRF continues to grapple with limitations stemming from the non-intuitive nature and limited controllability of implicit representations, along with hurdles in achieving high-resolution real-time rendering. Qiuxian Li et al. [23] proposed E2DSNeRF, which utilizes a high-dynamic-range and low-latency event camera to replace the conventional RGB-D camera as the input for NeRF. This approach enhances the rendering quality of sparse views, eliminating blur and ghosting artifacts caused by dynamic inputs. After NeRF, NeuS [24], and VolSDF [25] attempted to represent scenes using implicit representation methods and introduced neural volume rendering, making it possible to perform multi-view 3D reconstruction based on volume rendering. Kristina Prokopetc et al. [26] and Yujie Lu et al. [27] respectively, illustrate the significant advantages of deep learning-based multi-view 3D reconstruction methods by introducing their practical applications in medical mixed reality and civil engineering, transitioning from classical methods to deep learning approaches. These methods have become the current mainstream technology. As demonstrated by Sixu Li et al. [28], applying neural implicit 3D reconstruction algorithms to virtual/augmented reality enables instant 3D reconstruction with a time consumption of no more than 5 s per scene. Valeria Croce et al. [29] proposed combining neural implicit 3D reconstruction algorithms with actual photogrammetry for application in digital cultural heritage. Yingwei Ge et al. [30] utilized a 3D reconstruction algorithm that integrates Signed Distance Functions (SDF) with neural radiance fields for the reconstruction of ancient architectures, providing high-quality models of ancient buildings for surveying, mapping, visualization, and heritage preservation.

Despite its popularity, NeRF algorithm typically requires clear, static images to function effectively, leading to reduced performance when dealing with real-world scenarios that present non-ideal conditions such as complex reflections, low dynamic range, dark scenes, and blurriness resulting from camera motion or defocus photography [31]. The emergence of 3DGS (3DGS) technology presents a compelling solution to the previously encountered challenges posed by NeRF. The emergence of 3DGS (3DGS) has greatly accelerated rendering in novel view synthesis [32]. This technology incorporates a swift rendering algorithm capable of anisotropic splatting, which not only accelerates the training process but also augments rendering efficiency. Consequently, 3DGS has ushered in a paradigm shift within the realm of three-dimensional modeling. By representing scenes through explicit point clouds and leveraging a highly parallelized and differentiable rasterization pipeline, 3DGS achieves a notable improvement in both training speed and rendering efficiency, while simultaneously preserving the superior quality attributes of volume rendering. Furthermore, it creates an environment conducive to enhanced scene control.

2. Methods

2.1. 3D Rendering Algorithms-NeRF(Instant-NGP)

The NeRF (Neural Radiance Fields) observes the object located at a specific three-dimensional coordinate and from a particular viewpoint, estimating the color information and volume density through an MLP (Multi-Layer Perceptron) network. In this example, the direction is represented by a unit vector in the Cartesian coordinate system. o denotes the origin point from which the object is observed, and the camera ray is a virtual laser beam that emanates from this origin and projects towards the object, as depicted in Equation (1). This signifies that we are considering a point located along the vector, as seen from the origin.

$$r(t) = o + td. \tag{1}$$

2.1.1. Volume Rendering

The volume rendering process is a methodology that derives the expected color values that can appear in a pixel by utilizing the information obtained through the MLP (Multi-Layer Perceptron). This can be represented mathematically as shown in Equation (2).

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t)\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})dt, \tag{2}$$

t_n and t_f represent the depth of the 3D space to be rendered, while $\sigma(\boldsymbol{r}(t))$,$\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})$, and $T(t)$ refer to the density, color value, and transmittance, respectively, which are all derived from the MLP. In this scenario, c is associated with the density, such that as the density approaches 1, the transmittance decreases.

When light rays calculate their interaction with each point, the computation can include unnecessary pixels, limiting performance. To obtain sampling points, stratified sampling is used, which divides the interval between t_n and t_f into $\boldsymbol{N}$ strata, and randomly selects one sample from each stratum. The entire process can be represented by Equation (4) and applied to Equation (3).

$$T(t) = \exp(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))ds). \tag{3}$$

$$t_i \sim u[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_i)],. \tag{4}$$

2.1.2. Multi-Resolution Hash Encoding

A significant reduction in learning time is achieved by employing a multi-resolution hash encoding approach, as opposed to traditional position encoding methods. From an object-centric perspective, a large number of sampling points are placed along virtual rays, as illustrated in Figure 1.

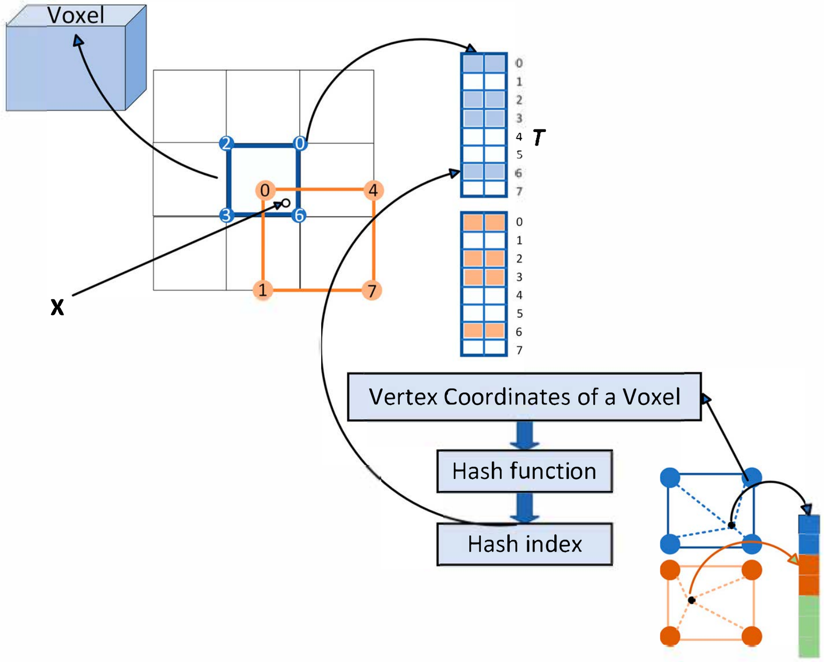

Figure 1. Diagram of Multi-Resolution Hash Encoding method. The blue, brown, orange, and green boxes represent the index calculation of hash tables at different Levels. Each Level has a different grid resolution. T denotes the size of the hash table.

Sampling points are encapsulated within voxel cubes located in 3D space, and the coordinates of these voxels are input into a hash function. In this example, the hash function is given by Equation (5). Here, $h(x)$ represents the hash index, and T is the limitation on the size of the hash index due to the operation denoted by mod.

$$h(x) = \left((\bigoplus_{i=1}^{d} x_i \pi_i) \right) \mathrm{mod} T. \quad (5)$$

By utilizing all 2D features matched through hash indices computed for each voxel vertex along each ray, distances between sampling points and voxel vertices are weighted to generate a single 2D feature. This feature is then input into the neural network through linear interpolation operations. At this point, the number of features that can be generated by a ray is calculated as the product of the pixel size in the image and the number of sampling points along the ray. The features are derived from voxels consisting of 16 different levels of sizes, all of which are concatenated, completing the multi-resolution hash encoding process.

In addition to employing the multi-resolution hash encoding method, the Tiny-Cuda-NN library based on Cuda and C++, along with a simple MLP for dimensionality reduction at each layer of the neural network, are utilized to reduce training and rendering time. By adopting the described methodology, excellent memory utilization has been achieved, and good results can be produced with minimal learning time.

2.2. 3D Rendering Algorithms-3DGS

The Three-Dimensional Gaussian Splatting (3DGS) technique defines the radiance field in a 3D scene on a discrete cloud of 3D Gaussian points to enable differentiable volume rendering. The technical process is illustrated in Figure 2. In 3DGS, each point is represented as an independent 3D Gaussian distribution [33], which can be mathematically expressed as Equation (6) below:

$$G(\boldsymbol{x}) = \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \sum (\boldsymbol{x}-\boldsymbol{\mu})). \quad (6)$$

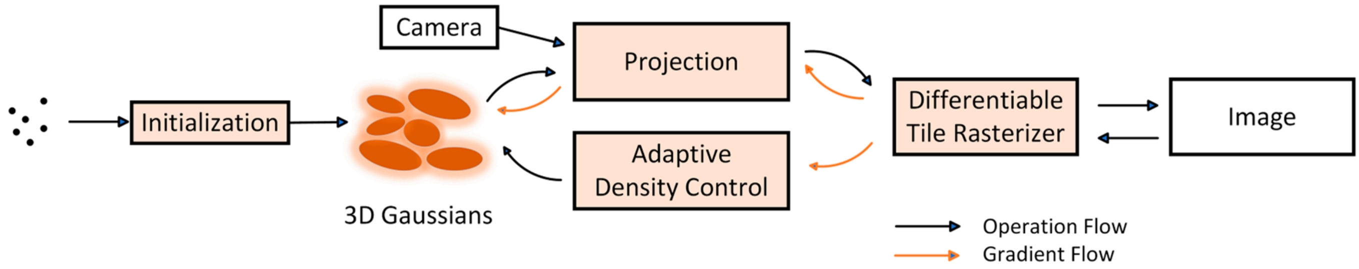

Figure 2. Overview of 3DGS technology.

In this context, $\boldsymbol{\mu}$ and $\sum$ represent the mean and covariance matrix, respectively, of the 3D Gaussian distribution. Each Gaussian point $\boldsymbol{P}$ is also assigned an opacity o and a color value $\boldsymbol{c}$, which together represent the radiance field of the 3D scene. Due to phenomena such as specular reflection and highlights, the color of the same object can vary depending on the viewing angle. To simulate the variation of color values concerning the viewing direction, 3DGS employs Spherical Harmonics (SHs) for modeling.

2.3. Evaluation Indices

Using PSNR (Peak Signal-to-Noise Ratio) [34], SSIM (Structural Similarity Index Measure) [35], and LPIPS (Learned Perceptual Image Patch Similarity) [36], the quality of the rendered images was evaluated.

2.3.1. PSNR

PSNR serves to indicate the lossy information associated with the generated image, as shown in Equation (7). Here, 'MAX' represents the maximum possible pixel value, and 'MAX' denotes the Mean Squared Error (MSE), which is the difference between each pixel, averaged across all pixels.

$$PSNR = 10 \log \frac{MAX^2}{MSE}. \tag{7}$$

2.3.2. SSIM

Unlike PSNR, SSIM is an evaluation method that indicates the image quality as perceived by humans, rather than simply numerical errors. It is calculated by assessing the brightness, contrast, and structure between images, as shown in Equation (8). In this context, μ_x and μ_y represent the average pixel values of the images, σ_x and σ_y denote the standard deviations of the pixel values, σ_{xy} is the covariance between the pixel values of the images, C_1 equals 6.5025, and C_2 equals 58.5225.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{({\mu_x}^2 + {\mu_y}^2 + C_1)({\sigma_x}^2 + {\sigma_y}^2 + C_2)}. \tag{8}$$

2.3.3. LPIPS

LPIPS is a model that employs AlexNet, VGG, and SqueezeNet to evaluate image similarity in a manner that mimics human perceptual characteristics. By leveraging intermediate layer features from networks trained on the ImageNet dataset, it measures the similarity between two features. The LPIPS calculation for input images is given by Equation (9), where l represents the layer, y and y_0 are unit-normalized feature vectors at the channel level, ω is a scaling factor, and H and W denote the height and width, respectively. Unlike PSNR and SSIM, LPIPS measures the distance between the feature vectors of the original and generated images, and thus, a lower value indicates better performance.

$$LPLIPS(x, x_0) = \sum_l \frac{1}{H_l W_l} \left\| \omega_l \odot (\hat{y'}_{h\omega} - \hat{y'}_{0h\omega}) \right\|_2^2. \tag{9}$$

2.4. Test Data Preparation

To enable a comprehensive and detailed evaluation of the algorithm, the corridor located on the fourth floor of the School of Aerospace Surveying and Mapping at Anhui University of Science and Technology has been designated as the testing environment, as depicted in Figure 3. This corridor, measuring a total length of 15 m, exhibits a highly diverse environmental design. It incorporates hanging decorations, a meticulously organized arrangement of tables and chairs, pristine white walls, transparent glass structures, and prominent corner elements. Collectively, these features create a testing space that is both unique and broadly representative.

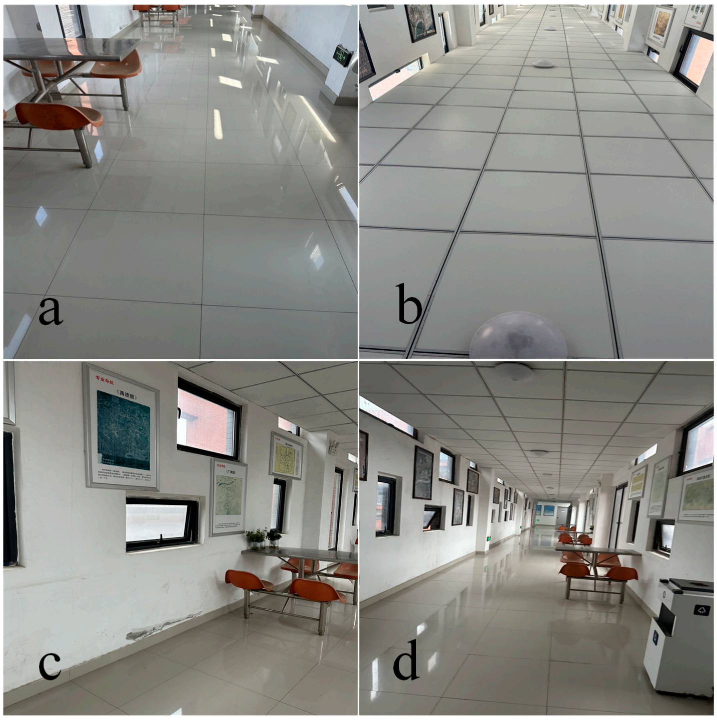

Figure 3. Study Area. (**a**) Floor; (**b**) Ceiling; (**c**) Table; (**d**) Full View.

To ensure the fairness and comprehensiveness of the evaluation, all experiments conducted in this study were performed in a standardized hardware environment, utilizing RTX 4060 graphics cards and 16 GB of RAM. This approach was implemented to mitigate any potential effects of hardware discrepancies on the experimental outcomes. The primary objective of these experiments is to accurately assess and optimize the performance of the algorithms.

3. Results

This study investigates the digital reconstruction of indoor scenes through the application of four distinct technical approaches, encompassing both traditional and advanced algorithms, to thoroughly examine efficient and precise reconstruction methodologies. Initially, we employed the industry-leading Context Capture Center Master (CC), utilizing its tilt photography technology specifically optimized for architectural applications to achieve high-quality reconstruction and rendering outcomes. Furthermore, to enhance our analytical framework and facilitate a more comprehensive comparative assessment, we also evaluated the modeling efficacy of the open-source COLMAP software (V3.7). Subsequently, in our pursuit of increased efficiency and speed in reconstruction and rendering processes, we incorporated NeRF technology, particularly the Instant-NGP method introduced by Thomas et al. [24]. Despite the notable advancements NeRF has achieved in the realm of reconstruction, it continues to encounter challenges related to real-time rendering and overall quality. Finally, we implemented the 3DGS algorithm for real-time rendering. As a recent advancement in the domain of differentiable point cloud rendering, 3DGS distinguishes itself from other technologies due to its substantial enhancements in speed, rendering efficiency, and accuracy. Specifically, in terms of optimization speed, rendering speed, and accuracy, 3DGS significantly surpasses NeRF technology, thereby elevating the capabilities of differentiable rendering technology to new heights.

3.1. Image Data Acquisition and Preprocessing

Due to the complex geometric features of the experimental area, significant emphasis was placed on managing image overlap at surface junctions and corners. By increasing the number of photographs captured, the overlap rate in these critical areas was maintained between 70% and 80%, thereby improving the effectiveness of multi-view image matching (see Figure 4). To address the challenge of modeling white walls, which frequently exhibit a lack of feature points, a technique involving the application of patterns on A4 paper, followed by their attachment to the walls, was utilized to artificially introduce feature points. This method effectively reduced modeling deficiencies and substantially improved both the accuracy and quality of the resultant models.

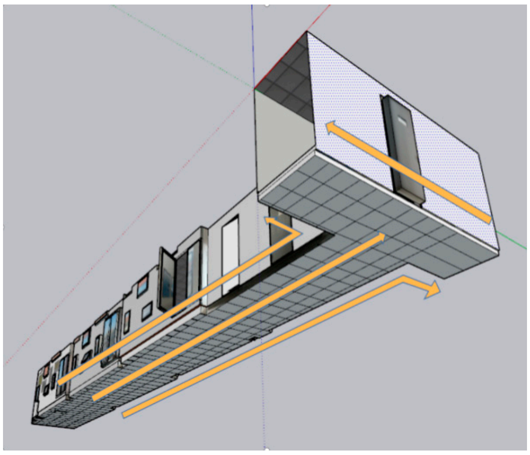

Figure 4. Schematic diagram of data collection. The arrows indicate the direction and trajectory of the shooting.

The image acquisition process involved the use of mobile phones to capture photographs, which were subsequently subjected to a comprehensive preprocessing protocol. This protocol aimed to eliminate images that exhibited high exposure, reflections, and other defects, thereby ensuring the integrity of the input data and establishing a solid foundation for subsequent modeling and rendering tasks. The preprocessing workflow not only streamlined the data processing procedure but also significantly enhanced the overall quality and detailed representation of the final three-dimensional model. This improvement provides considerable support for the precise reconstruction and rendering of indoor environments.

3.2. Algorithm Performance Comparison

In the designated study area, a cohesive collection of 170 high-quality photographs was assembled to serve as the foundational data for modeling purposes. Four distinct technical methodologies—specifically, Structure from Motion (CC), COLMAP, NeRF, and 3DGS—were employed to independently generate three-dimensional models. A comprehensive evaluation was subsequently conducted through comparative analysis.

3.2.1. Comparison of Rendering Time and Results Among Four Algorithms

During the preliminary exploration phase, significant technical difficulties emerged when utilizing CC to process complex scenes characterized by extensive areas of white walls and floors. Specifically, conventional three-dimensional reconstruction algorithms encountered challenges in extracting an adequate number of feature points from these expansive white surfaces, which resulted in unsuccessful feature matching. This limitation subsequently hindered precise registration during the aerial triangulation phase, ultimately leading to unsuccessful modeling outcomes. To mitigate this issue and achieve a comprehensive model, a substantial increase in the dataset was considered, resulting in the incorporation of 895 unrestricted photographs for re-modeling through CC, as illustrated in Figure 5a. Although initial modeling results were obtained, the output displayed significant voids and protrusions, and manual remediation proved to be inefficient. Consequently, OSketch Up software(V1.0) was employed, leveraging its interactive object-based modeling capabilities to enhance the preliminary CC model, as depicted in Figure 6c,d. This integration significantly improved processing efficiency and effectively reduced the time required for manual corrections. Ultimately, by combining the functionalities of both software tools, a successful object-based construction of the model was realized, with the final result presented in Figure 6b, thereby accomplishing both efficient and accurate three-dimensional model reconstruction.

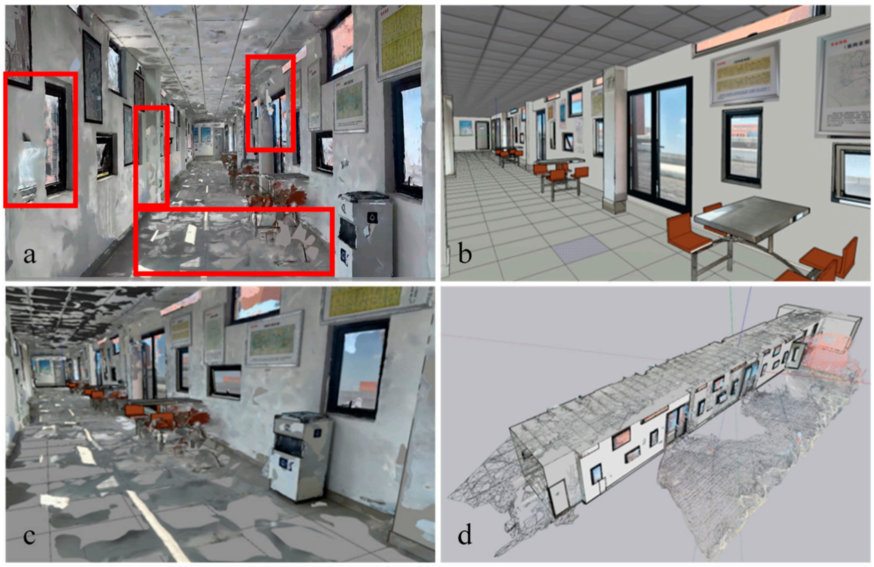

Figure 5. Model construction and rendered image generation. (**a**) CC modeling diagram, the red border highlights the damaged area of the model.; (**b**) OSketch Up individualized rendering; (**c**,**d**) OSketch Up interactive operation diagrams.

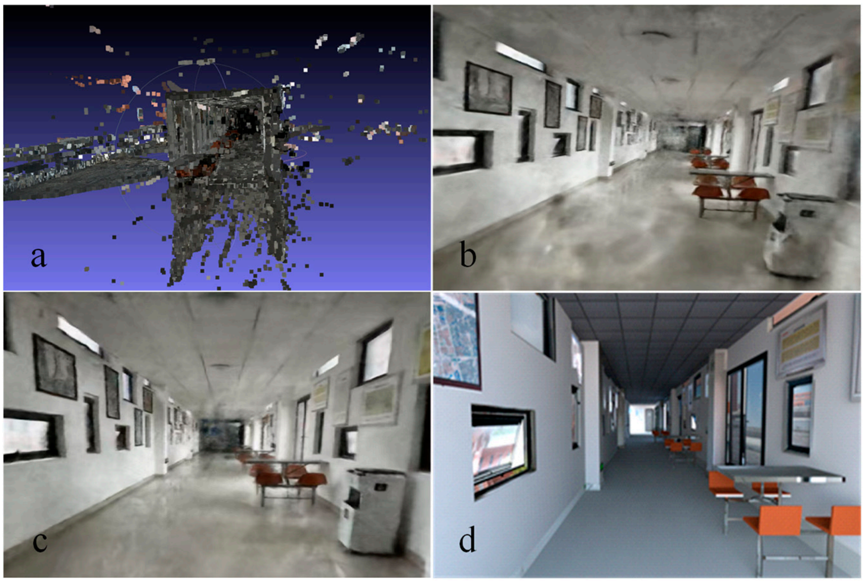

Figure 6. Comparison Chart of Rendering Effects Among Different Algorithms: (**a**) COLMAP; (**b**) NeRF (Instant-NGP); (**c**) 3DGS; (**d**) CC.

In the comprehensive evaluation of the effectiveness of four three-dimensional reconstruction and rendering methodologies—CC, COLMAP, NeRF, and 3DGS—stringent assessment criteria were employed. Given CC's limitations in accurately modeling certain complex scenes, an augmented dataset was utilized to enhance contrast for comparative analysis. For the assessment of COLMAP, NeRF, and 3DGS, a standardized dataset consisting of 170 images was meticulously curated to ensure fairness and comparability of the results. As illustrated in Figure 6, the upper row presents the rendering effect comparisons using 170 images, while the lower row displays the rendering effects derived from 895 images. Through extensive visual analyses, the following conclusions were drawn:

Utilizing 170 images, COLMAP, a conventional three-dimensional reconstruction tool, produced a model with numerous voids and incomplete features. In contrast, with the application of 895 images, after CC modeling, object-based modeling through OSketch Up, and subsequent rendering processes, the resulting model appeared relatively complete and visually appealing. However, in terms of color fidelity, the model generated by CC exhibited slightly inferior to those produced by NeRF and 3DGS.

Both NeRF and 3DGS demonstrate exceptional performance in terms of model completeness, realism, and overall quality, attributable to their respective technical advantages. NeRF, utilizing its sophisticated neural network architecture, adeptly captures and reproduces intricate details and lighting variations within a scene, thereby facilitating the generation of a highly realistic three-dimensional environment. Conversely, 3DGS exhibits distinct strengths in managing complex geometric configurations and enhancing scene consistency, resulting in rendered images that closely resemble real-world environments.

The initial methodology employed, which involved modeling with CC, followed by individualization through OSketch Up and subsequent rendering, required a duration of five days. In contrast, the modeling process utilizing COLMAP was accomplished in only six hours. It is noteworthy that the NeRF and 3DGS algorithms exhibited exceptional efficiency, achieving image rendering within a matter of minutes, thus positioning themselves as the most effective techniques in this context.

3.2.2. Comparison of Reconstruction Results with Varying Image Quantities

Comparative analyses of the reconstructions within the experimental area were performed utilizing 60, 110, and 170 images, with the results of these comparisons illustrated in Figure 7. In this figure, 'n' denotes the number of images utilized.

As depicted in Figure 7, the three-dimensional reconstruction algorithms NeRF and 3DGS demonstrate considerable superiority over COLMAP. An increase in the number of input images indicates that both NeRF and 3DGS effectively manage large-scale datasets, demonstrating a clear trend toward improved modeling quality. This observation underscores the robustness and advancements of these two algorithms in the field of three-dimensional reconstruction. However, it is noteworthy that when assessing the three-dimensional reconstruction and rendering results using a dataset comprising 170 images, 3DGS did not achieve the performance of NeRF and, in certain respects, lagged behind renderings generated from a smaller number of images. This discrepancy invites further investigation, with the underlying issue potentially related to the quality of the initial point cloud data.

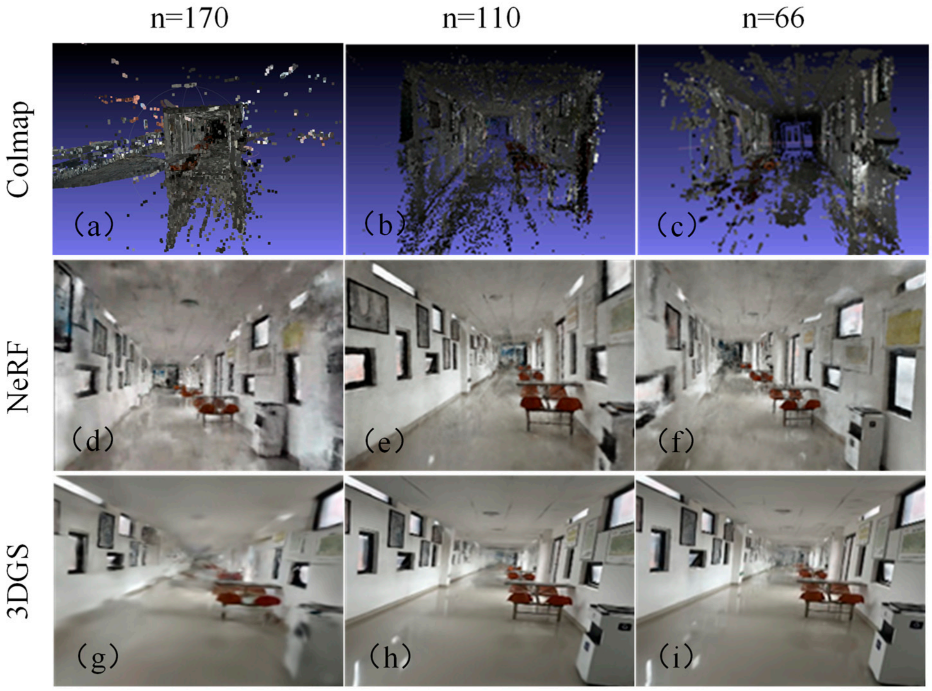

Figure 7. A comparison of reconstruction results between COLMAP, NeRF, and 3DGS methods under varying numbers of remote sensing images. (**a**–**c**) Showcase the modeling effects of COLMAP when the number of images is 170, 110, and 66, respectively; (**d**–**f**) Demonstrate the modeling outcomes of NeRF (Instant-NGP) with 170, 110, and 66 images; (**g**–**i**) Present the modeling performance of 3DGS for the same sets of 170, 110, and 66 images.

Specifically, the efficacy of the 3DGS method is heavily contingent upon the quality and accuracy of the initial point cloud data. In the data preprocessing phase, the failure to adequately eliminate redundant Gaussian points can result in noise, which negatively impacts the subsequent rendering process. Notably, covariance—a fundamental characteristic of Gaussian points—is particularly vulnerable to noise interference during its calculation and representation, which can lead to a decline in image quality during the rendering phase.

At critical junctures involving 66 and 110 images, while NeRF and 3DGS demonstrate a significant degree of consistency in the macro-level rendering of the overall scene, notable differences emerge at the micro-level.

3.2.3. Comprehensive Evaluation of 3DGS Performance

Two representative facets of the models were deliberately selected for an in-depth comparison, to clarify the nuanced differences between the two algorithms in terms of texture detail processing, geometric structure restoration, and lighting simulation. This meticulous methodology was implemented across datasets comprising 66 and 110 images.

The experimental results presented in Figures 8 and 9 indicate that contrary to initial anticipation, the rendering quality of the NeRF algorithm did not improve progressively with an increase in the number of images; instead, it demonstrated a slight decline. This finding hints at potential limitations within the NeRF algorithm when managing datasets of specific sizes, or it may reflect the intricate impacts of data augmentation on model optimization. Conversely, the rendering quality of the 3DGS algorithm remained relatively stable as the number of images increased from 66 to 110, with no significant changes detected. This finding highlights the superior adaptability and stability of the 3DGS algorithm when processing datasets of varying sizes, thereby ensuring consistent and high-quality rendering outcomes.

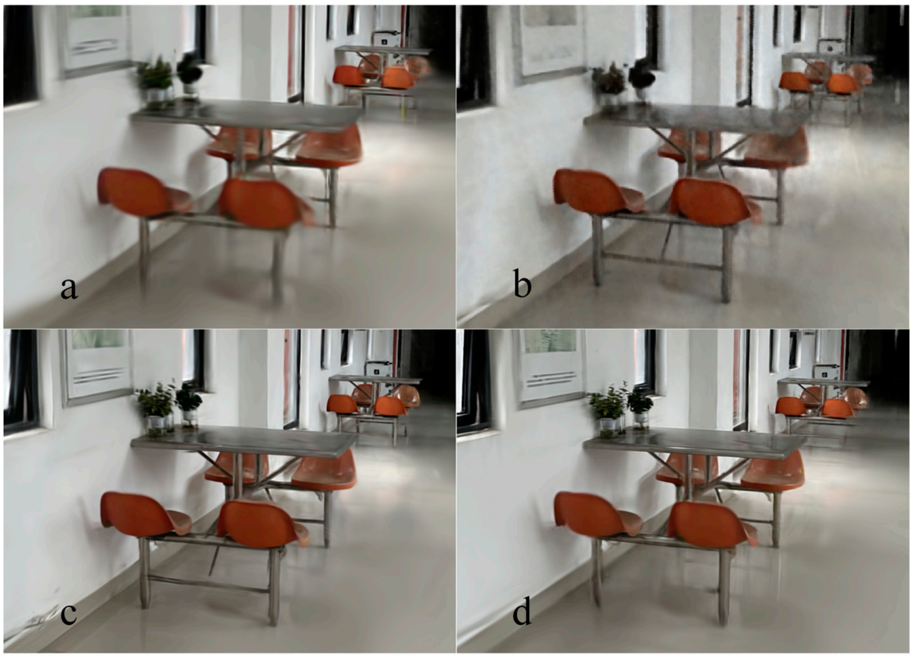

Figure 8. A comparative analysis of rendering outcomes of tables/chairs. (**a**,**b**) Modeling accuracy of chairs utilizing NeRF (Instant-NGP) with 110 and 66 Images, respectively. (**c**,**d**) Reconstruction fidelity of chairs by 3DGS for 110 and 66 images, respectively.

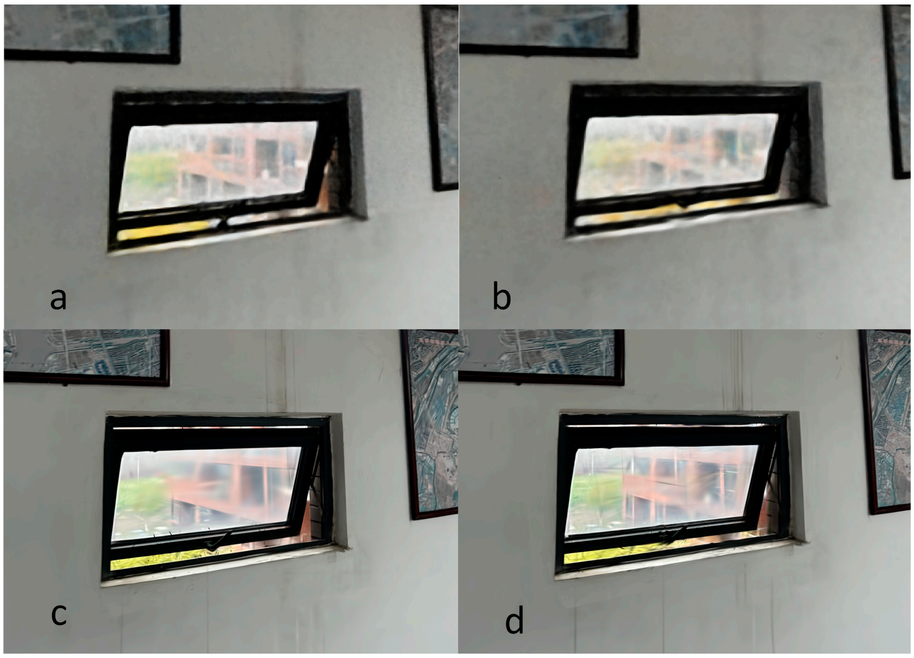

Figure 9. A comparative analysis of rendering outcomes of windows. (**a**,**b**) Modeling performance of windows achieved by NeRF (Instant-NGP) with 110 and 66 images, respectively. (**c**,**d**) Reconstruction precision of windows via 3DGS for the corresponding sets of 110 and 66 images, respectively.

At the level of overall perception, a meticulous comparison of images rendered using the two techniques, 3DGS and NeRF, clearly demonstrates that, under equivalent conditions, images produced by 3DGS exhibit superior clarity and quality in terms of detail representation. When closely examining the finer elements within these images, the details rendered by 3DGS appear richer and sharper. Conversely, while NeRF can generate high-quality images, it falls slightly short in terms of the completeness and clarity of certain details, failing to match the level of detail achieved by 3DGS.

It is worth noting that, despite an increase in photos from 66 to 110, the 3DGS algorithm maintains a running time difference of less than ten minutes when processing these two datasets of different sizes. This demonstrates the algorithm's efficiency in handling datasets of varying sizes. However, upon in-depth analysis of various performance indicators, it was observed that the dataset with 66 images exhibited superior LPIPS, PSNR, and SSIM values during the iterative process. This finding suggests that, under specific algorithm and iteration conditions, rendering quality is not always positively correlated with the number of photos. In other words, an increase in data volume does not necessarily translate to an improvement in rendering quality and may, in some instances, lead to a decline in performance indicators. Therefore, in practical applications, it is crucial to reasonably control the dataset size based on the characteristics of the specific algorithm and iteration conditions to achieve optimal rendering results.

To accurately assess the performance of the 3DGS algorithm, image datasets of different sizes (110 and 66 images) were subjected to 7 k, 15 k, and 60 k iterations, and key performance indicators such as LPIPS (Learned Perceptual Image Patch Similarity), PSNR (Peak Signal-to-Noise Ratio), and SSIM (Structural Similarity Index) were recorded, as shown in Table 1. As the number of iterations increased, the LPIPS values gradually approached zero, indicating that the rendered images generated by the 3DGS algorithm became increasingly perceptually similar to real-world scenes, thereby significantly improving the rendering quality. Simultaneously, the increase in PSNR values further reflected the improved quality of the rendered images, specifically manifested as a reduction in image distortion. This trend was consistent with the changes in LPIPS values, collectively validating the optimization effect of the 3DGS algorithm during the iteration process. Additionally, a continuous rise in SSIM values was observed, indicating that the similarity between the rendered images and real-world scenes at the structural level was also continuously enhanced. This finding further underscores the stability and reliability of the 3DGS algorithm in preserving scene structural information. In summary, the changes in these performance indicators not only reveal the continuous optimization of algorithm performance but also provide a more comprehensive and in-depth basis for evaluating and understanding the algorithm's capabilities.

Table 1. Comparative indices of 3DGS performance between 110 and 66 images.

Number of Images (Sheets).	Iteration Count.	Times (min)	LPIPS	PSNR	SSIM
110	7 k	12	0.360	22.874	0.802
	15 k	32	0.275	25.418	0.844
	60 k	350	0.170	29.145	0.898
66	7 k	12	0.369	22.708	0.796
	15 k	39	0.260	25.778	0.852
	60 k	340	0.137	30.451	0.915

3.3. Parameter Tuning

Despite the outstanding performance of the 3DGS algorithm in indoor modeling and rendering, it still exhibits deficiencies when examined through panoramic images, as illustrated in Figure 10. During the processing of high-resolution original images (3024 × 4032), the algorithm inevitably introduces a certain degree of information loss due to the step of resizing the images to 1.6 k resolution, which in turn affects the rendering quality of areas such as ceilings. Furthermore, although adaptive control of the Gaussian function can effectively represent scenes ranging from sparse to dense, balancing the issues of "under-reconstruction" and "over-reconstruction" is required when filling in blank areas. This balancing process is highly sensitive to the selection of hyperparameters for thresholds and density levels, which directly impacts the algorithm's runtime speed and the final rendering quality. Additionally, the learning rate (lr), a key factor in adjusting model parameters in deep learning, has a direct influence on covariance training and the quality of the Gaussian model. Therefore, to optimize the model's performance, fine-tuning the learning rate is necessary to ensure it functions optimally during the training process. For parameter selection, we start with the original parameter settings of the algorithm as a baseline. We then adjust one parameter value at a time, continually tweaking its magnitude, until the PSNR, SSIM, and LPIPS values all perform well or even surpass the original algorithm's effectiveness. Based on this process, we select the parameter that has the greatest impact on these three metrics. Finally, we incorporate the optimal parameter values obtained from the aforementioned experiments into the algorithm to generate qualitative and quantitative results, thereby achieving the goal of parameter tuning.

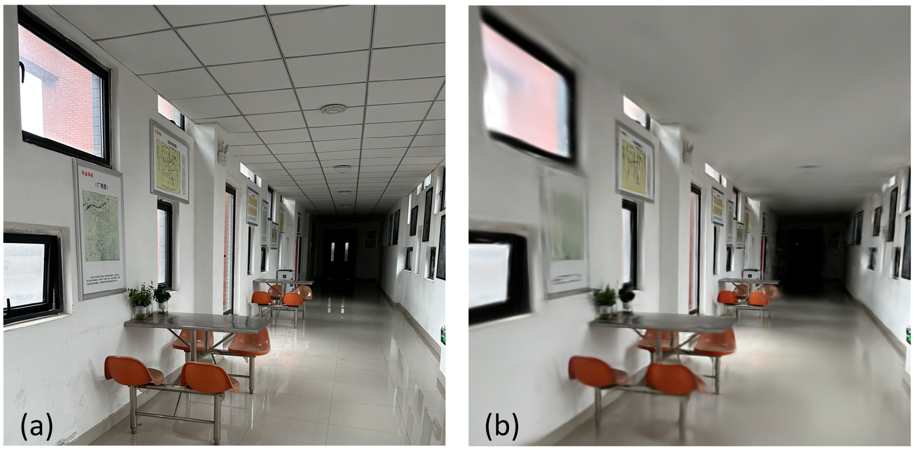

Figure 10. Comparison of scene images under different numbers of best pictures and iterations. (**a**) Original image; (**b**) rendered image.

In summary, despite the 3DGS algorithm's excellence in multiple aspects, attention should be paid to its potential limitations in practical applications, and further enhancements in performance can be achieved through precise parameter tuning.

3.3.1. Comparison of the Effects of the 3DGS Algorithm on Images of Different Resolutions

Under the conditions of fixing the number of photos at 66 and the number of iterations at 7000, we conducted a comparison of the training durations for images of varying resolutions. Additionally, we calculated the LPIPS, PSNR, and SSIM values, with the specific data presented in Table 2. The experimental outcomes reveal a clear trend: as the resolution of the image's decreases, the LPIPS value progressively approaches zero, the PSNR value steadily increases, and the SSIM value initially declines but then rises, gradually converging towards 1. This trend indicates that the training effectiveness gradually enhances with decreasing resolution. However, it is crucial to recognize that reducing resolution essentially entails down-sampling the image information, which inevitably results in a loss of image details. Therefore, it is not advisable to excessively decrease the resolution. Figure 11 shows a comparison of the dimensions of rendered images at various resolutions. After carefully balancing the performance indicators of LPIPS, PSNR, and SSIM, along with considerations of efficiency and the desired size of the rendered images, we determined that 0.8 k is the optimal resolution. This choice not only guarantees the rendering quality but also enhances training efficiency to a certain extent while mitigating the issue of excessive image information loss.

Table 2. Comparative analysis of various indices for the 3DGS algorithm at different resolutions.

Res	Times (min)	LPIPS	PSNR	SSIM
0.3 k	2	0.169	25.978	0.867
0.5 k	2	0.282	24.597	0.831
0.8 k	4	0.355	23.650	0.806
1.2 k	6	0.375	23.038	0.798
1.5 k	8	0.372	22.779	0.797
1.6 k	12	0.360	22.708	0.796
2 k	18	0.358	22.680	0.799
2.9 k	1662	0.334	22.530	0.806

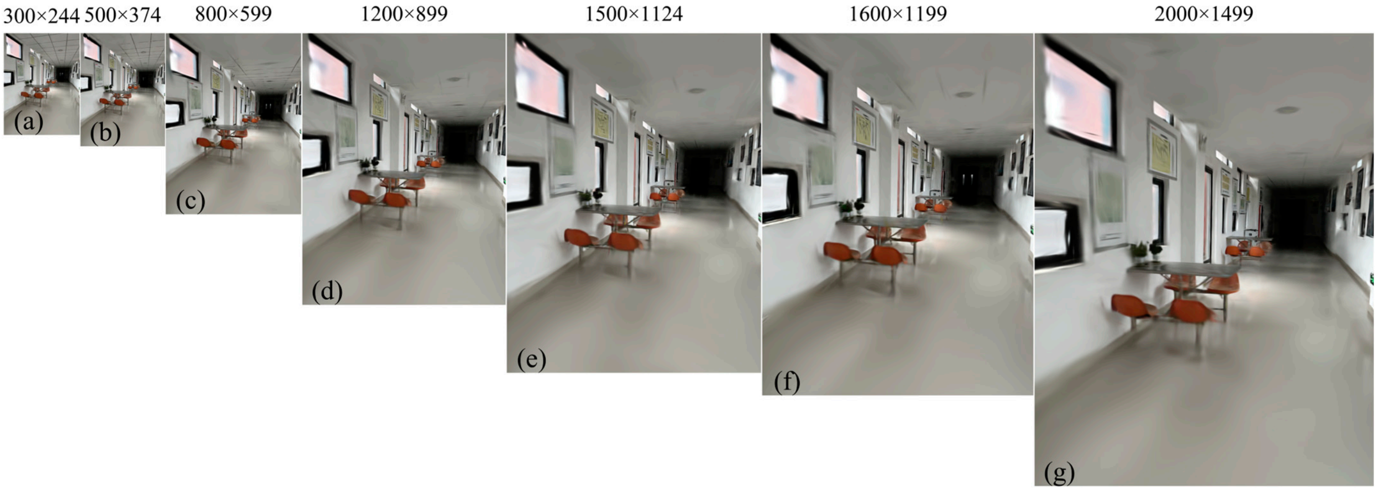

Figure 11. Comparison of dimensions for rendered images at various resolutions. (**a**–**g**) represent rendered images with resolutions of 0.3 k, 0.5 k, 0.8 k, 1.2 k, 1.5 k, 1.6 k, and 2 k, respectively.

3.3.2. Comparison of the Effects of the 3DGS Algorithm with Different Thresholds τ for the Average Magnitude of Spatial Position Gradients

As shown in Table 3, with the number of photos controlled at 66 and iterations set to 7000, we compared the training duration and calculated the LPIPS, PSNR, and SSIM values for different threshold values of the average magnitude of spatial position gradients. Figure 12 presents a qualitative comparison of the effects of the 3DGS algorithm under different threshold values for the average magnitude of spatial position gradients. Visually, there is no significant difference among the results obtained with different threshold values. However, from a quantitative perspective, when the threshold is set to 0.0003, the LPIPS value is closest to 0, the PSNR value is the highest, and the SSIM value is also the highest, with no significant difference in training duration. Therefore, selecting a threshold of 0.0003 is considered appropriate.

Table 3. Comparison of various indices of the 3DGS algorithm under different thresholds of gradient average magnitude in different spatial locations.

τ	Times (min)	PSNR	SSIM	LPIPS
0.0001	4	22.363	0.793	0.36
0.0002	4	22.708	0.796	0.36
0.0003	4	22.779	0.807	0.364
0.0004	3	22.633	0.795	0.369

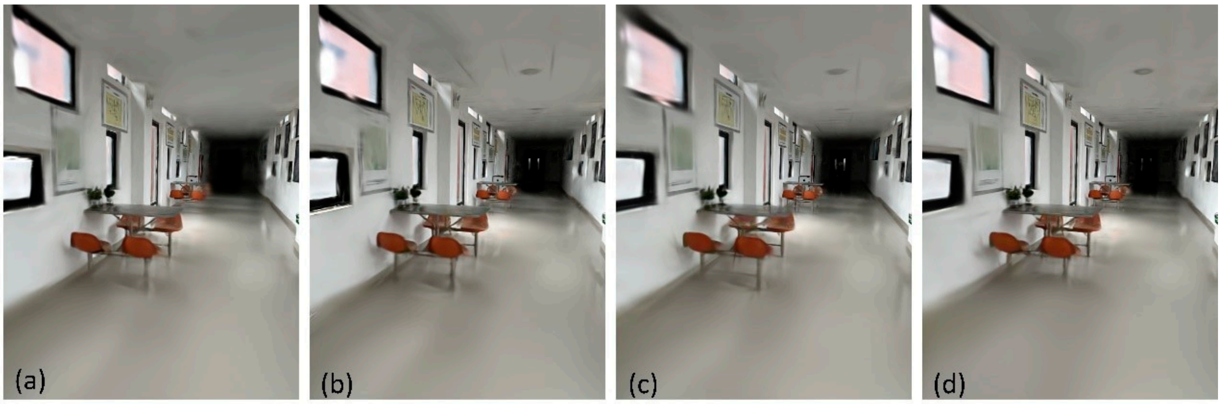

Figure 12. Comparison of rendering effects for different threshold values of the average magnitude of spatial position gradients. (**a**–**d**) represent the rendering effects when the threshold values are set to 0.0001, 0.0002, 0.0003, and 0.0004, respectively.

3.3.3. Comparison of the Effects of the 3DGS Algorithm Under Different Scaling Learning Rates (Scaling_lr)

As shown in Table 4, with the number of photos controlled at 66 and iterations set to 7000, we compared the training duration and calculated the LPIPS, PSNR, and SSIM values for different learning rates at various scaling scales. Figure 13 presents a visual comparison of the effects, and there is not much difference. However, as shown in Table 4, when the learning rate is set to 0.005, the performance of various indicators is the best, which aligns with the original algorithm's approach as discussed in this paper.

Table 4. Comparison of various indices of the 3DGS algorithm under different scaling learning rates (Scaling_lr).

Scaling_lr	Times (min)	LPIPS	PSNR	SSIM
0.004	10	0.358	22.77	0.796
0.005	10	0.36	22.708	0.796
0.006	10	0.354	22.582	0.796
0.008	9	0.351	22.432	0.796

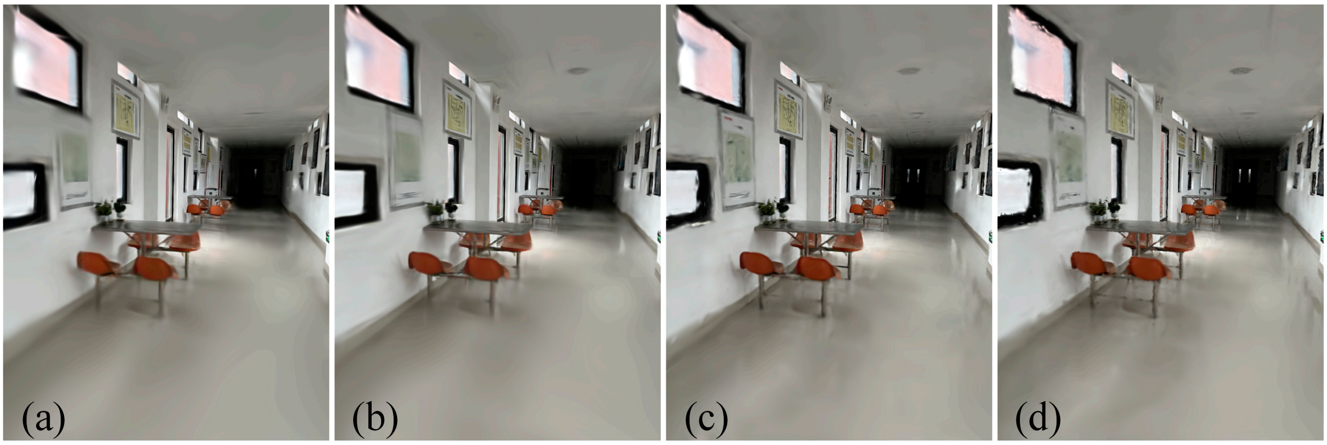

Figure 13. Comparison of rendering effects for different learning rates at various scaling scales. (**a**–**d**) represent the rendering effects when the scaling scale learning rates are set to 0.004, 0.005, 0.006, and 0.008, respectively.

3.3.4. Comparison of the Effects of the 3DGS Algorithm Under Different Hyperparameter Settings for Controlling the Density Level (p)

This parameter p represents the percentage of the density level of the Gaussian kernel. Different hyperparameter settings will result in varying degrees of density for the Gaussian kernel, which in turn affects the subsequent cloning and cropping of the Gaussian kernel. It serves as a condition for deciding whether to clone or crop the Gaussian kernel. When the maximum scaling of a Gaussian kernel in a certain direction exceeds a certain threshold, it will be cropped; otherwise, it will be cloned to increase the density of the Gaussian kernel. Figure 14 presents a visual comparison of the effects, and there is not much difference. According to Table 5, when the SSIM value does not change significantly and the time cost is comparable, the LPIPS and PSNR values perform best when the value of p is set to 0.001.

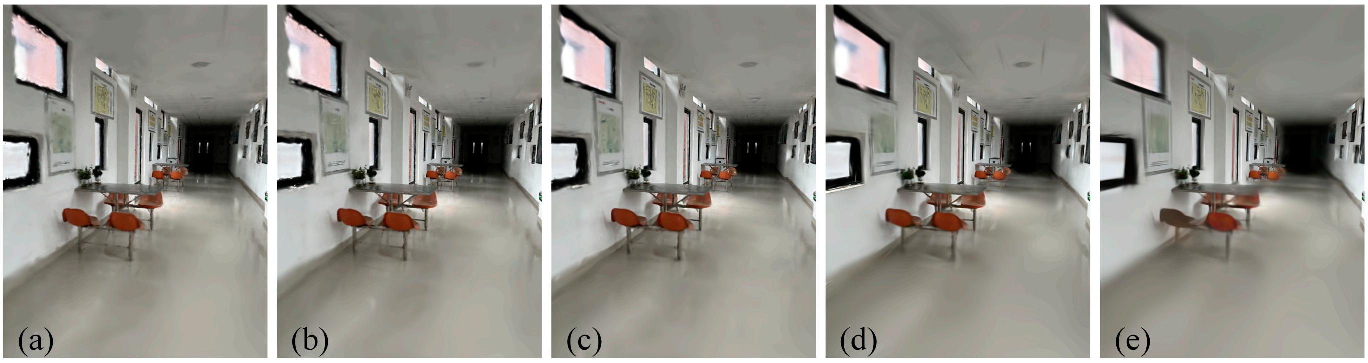

Figure 14. Comparison of rendering effects for different hyperparameter settings. (**a**–**e**) represents the rendering effects when the hyperparameter settings are 0.0005, 0.001, 0.002, 0.01, and 0.1, respectively.

Table 5. Comparison of various indices for the 3DGS algorithm under different hyperparameter settings for density levels (p).

p	Times (min)	LPIPS	PSNR	SSIM
0.0005	10	0.377	23.017	0.797
0.001	10	0.36	23.178	0.799
0.002	11	0.369	22.893	0.793
0.01	10	0.36	22.708	0.796
0.1	10	0.382	21.522	0.786

3.3.5. Ranking of Sensitivity for Four Parameters

By substituting the optimal values of the selected parameters into the algorithm and comparing them with the original results, we obtained the differences in Structural Similarity Index (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), and Peak Signal-to-Noise Ratio (PSNR) values for each parameter. This allowed us to rank the parameters based on their impact on the algorithm, preparing for the next step of optimization. As shown in Table 6, the resolution has the greatest impact on the algorithm: it increases SSIM and PSNR values while decreasing the LPIPS value, outperforming other parameters. Following that, the hyperparameter controlling the density of the Gaussian kernel and the threshold value of the average magnitude of spatial position gradients also have significant effects. The scaling scale learning rate has the least impact on the algorithm. Therefore, during the parameter tuning process, we will no longer alter the scaling scale learning rate.

Table 6. Sensitivity ranking table for four parameters.

Parameter	PSNR	SSIM	LPIPS
Res	+0.942	+0.01	−0.005
p	+0.47	+0.003	0
τ	+0.071	+0.011	+0.004
Scaling_lr	0	0	0

3.3.6. Optimization Results

After conducting the aforementioned experiments, the final determined algorithm parameters are as follows: a resolution of 0.8 k, a threshold set to 0.0003, and a hyperparameter controlling the density of the Gaussian kernel set to 0.001. With these parameter settings, the algorithm demonstrates the best performance in indoor rendering.

When using a self-collected dataset, Table 7 compares the three key image quality evaluation metrics—LPIPS, PSNR, and SSIM values—before and after algorithm tuning. The results show that the PSNR value increased by 4.3%, marking a significant enhancement in image sharpness or reconstruction quality. Additionally, the SSIM value also improved by 0.2%, indicating better preservation of image structural information. Figure 15 below shows a detailed comparison of image details before and after algorithm tuning. By observing the comparison images, it is clear that the post-tuning images exhibit sharper and clearer details, with better preservation of structural information. These changes directly reflect the positive impact of algorithm tuning on image quality.

Table 7. Quantitative comparison results before and after algorithm tuning.

p	τ	Res	Times (min)	LPIPS	PSNR	SSIM
0.001	0.003	0.8 k	10	0.36	23.694	0.798
0.01	0.002	1.6 k	10	0.36	22.708	0.796

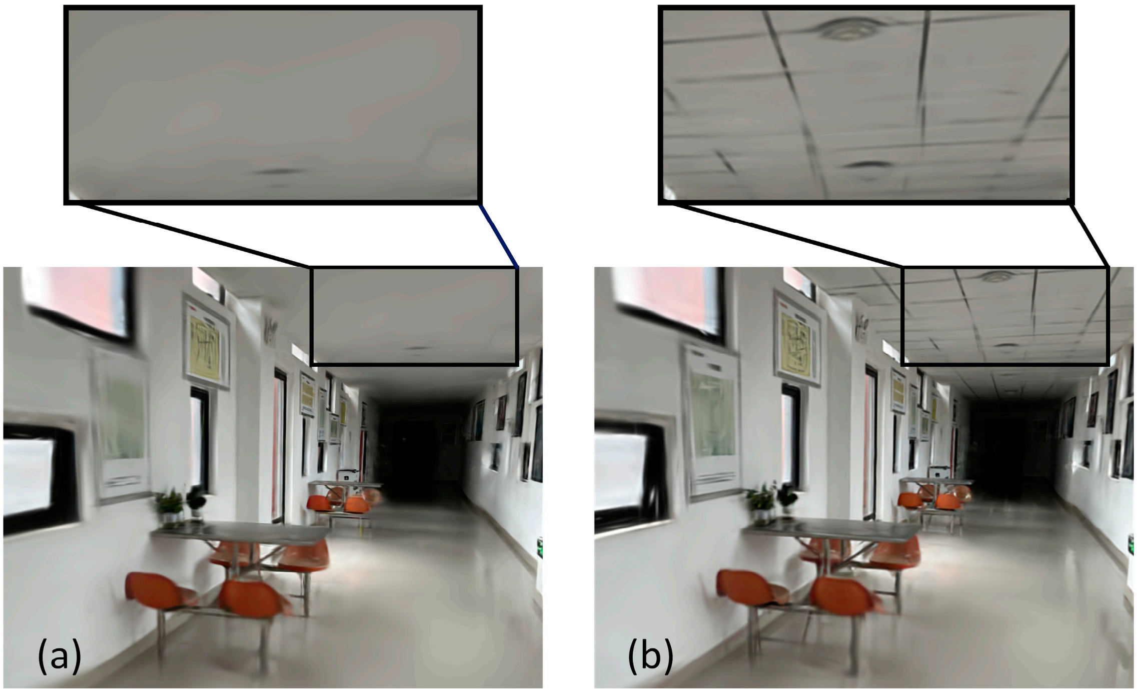

Figure 15. Comparison diagram of ceiling area before and after algorithm optimization. (**a**) Before optimization; (**b**) After optimization.

The learning outcomes displayed in Figure 16 provide further substantiation for the rapid convergence and stability of the refined 3DGS algorithm across iterations. As the iteration count progresses, the algorithm's learning performance exhibits a steady enhancement, with all performance metrics demonstrating favorable trends. This underscores the capability of the 3DGS algorithm to not only deliver high-quality rendering results on constrained datasets but also to iteratively refine its performance, thereby enhancing its adaptability to a wide range of application scenarios and demands.

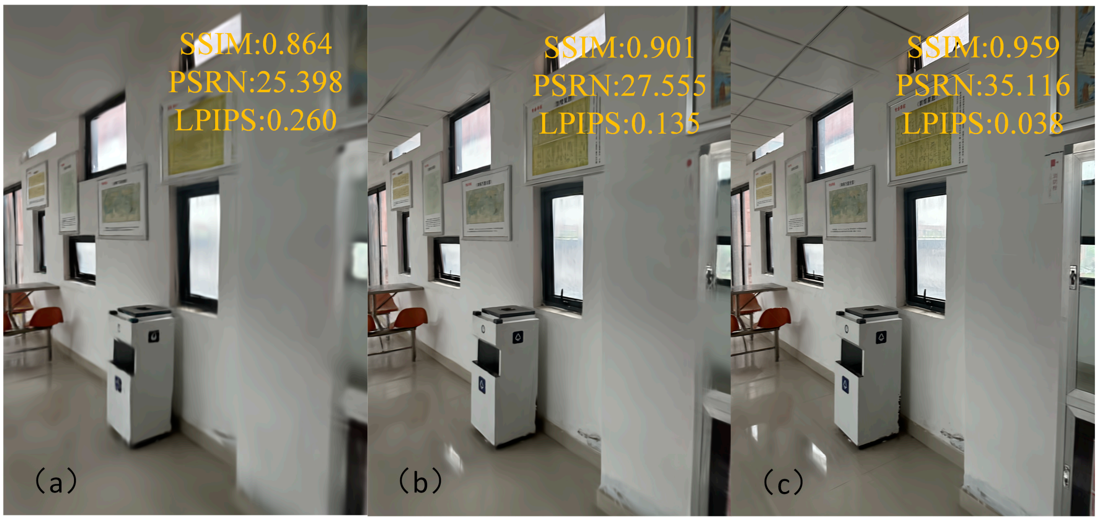

Figure 16. Iterative 3DGS training results. (**a**) Original image; (**b**) Training result after 7 k iterations; (**c**) Training result after 30 k iterations.

4. Discussion

We have conducted an extensive investigation into the potential applications of the 3DGS algorithm within the realm of indoor scene modeling. Through a series of meticulously designed iterative experiments and a rigorous performance evaluation framework, we have conducted a comprehensive performance comparison between this algorithm and the current state-of-the-art indoor scene modeling techniques, including CC (traditional computer vision methods), COLMAP (a dense reconstruction technique grounded in Structure from Motion), and NeRF (particularly its efficient variant, Instant-NGP). Building upon this comparative analysis, we have undertaken meticulous tuning of the 3DGS algorithm to further augment its adaptability and expressive capabilities in indoor scene modeling.

During the tuning phase, we paid particular attention to the algorithm's proficiency in detail preservation, rendering efficiency, and its handling of intricate indoor structures, such as ceilings and corners. By refining the adaptive control strategy of the Gaussian function, optimizing the image resolution processing pipeline, and finely adjusting crucial parameters like the learning rate, we successfully bolstered the algorithm's performance in capturing intricate indoor scene details, minimizing information loss, and accelerating the rendering process. These enhancements render the 3DGS algorithm more adept at constructing high-precision, high-quality indoor scene models, thereby providing robust technical support for subsequent applications in virtual reality, augmented reality, and indoor navigation.

While the 3DGS technique has made strides in indoor scene modeling, it still encounters inefficiencies and suboptimal performance when dealing with high-resolution images, which restricts its application in complex scenarios. Despite advancements in improving reconstruction efficiency and rendering quality, the technique remains constrained by inherent limitations and a significant reliance on auxiliary technologies, such as SFM.

Future research endeavors should concentrate on optimizing algorithm performance and mitigating the dependency on auxiliary technologies to achieve more efficient and precise 3D reconstruction, thereby broadening its application scope to encompass a broader range of indoor scenes.

Author Contributions: Xinjian Fang proposed the methodology, wrote the manuscript and was the corresponding author. Yingdan Zhang contributed to improving the methodology. Hao Tan, Chao Liu, and Xu Yang helped edit and improve the manuscript. Funding acquisition, Xu Yang and Xinjian Fang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 42304050); Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety (No. CSBD2022-ZD04).

Data Availability Statement: The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tran, H.; Khoshelham, K.; Kealy, A.; Díaz-Vilariño, L. Shape Grammar Approach to 3D Modeling of Indoor Environments Using Point Clouds. *J. Comput. Civ. Eng.* **2019**, *33*, 04018055. [CrossRef]
2. Naseer, M.; Khan, S.; Porikli, F. Indoor scene understanding in 2.5/3D for autonomous agents: A survey. *IEEE Access* **2018**, *7*, 1859–1887. [CrossRef]
3. Jacq, K.; Ployon, E.; Rapuc, W.; Claire, B.; Cécile, P.; Didier, C.; Fanget, B. Structure-from-motion, multi-view stereo photogrammetry applied to line-scan sediment core images. *J. Paleolimnol.* **2021**, *66*, 249–260. [CrossRef]
4. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Jonathan, T.B.; Ravi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* **2021**, *65*, 405–421. [CrossRef]
5. Kerbl, B.; Kopanas, G.; Leimkuehler, T.; Drettakis, G. 3DGS for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 139. [CrossRef]
6. Zhilu, Y.; You, L.; Shengjun, T.; Ming, L.; Renzhong, G.; Wang, W. A survey on indoor 3D modeling and applications via RGB-D devices. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 815–826.
7. Li, J.K. UAV Oblique Photography Model and Single-Dimensional Treatment. *J. Geol. Hazards Environ. Preserv.* **2020**, *31*, 95–99.
8. Ye, J.; Yan, C.; Jianing, Q. Interior Soft Decoration Product Design Based on 3D Modeling and Image Processing Technology. *Mob. Inf. Syst.* **2022**, *2022*, 9095614.
9. Zavar, H.; Arefi, H.; Malihi, S.; Maboudi, M. Topology-Aware 3D Modelling of Indoor Spaces from Point Clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 267–274. [CrossRef]
10. Wang, J.; Shoudong, H.; Liang, Z.; Janet, G.; Sean, H.; Chengqi, Z.; Wang, X. High quality 3D reconstruction of indoor environments using RGB-D sensors. In Proceedings of the 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), Siem Reap, Cambodia, 18–20 June 2017; pp. 1739–1744. [CrossRef]
11. Cui, Y.; Li, Q.; Yang, B.; Wen, X.; Chi, C.; Dong, Z. Automatic 3-D Reconstruction of Indoor Environment With Mobile Laser Scanning Point Clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3117–3130. [CrossRef]
12. Liu, J.; Luo, J.; Hou, J.; Danqi, W.; Guoqiang, F.; Zhang, X. A BIM Based Hybrid 3D Indoor Map Model for Indoor Positioning and Navigation. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 747. [CrossRef]
13. Poux, F.; Neuville, R.; Nys, G.-A.; Billen, R. 3D Point Cloud Semantic Modelling: Integrated Framework for Indoor Spaces and Furniture. *Remote Sens.* **2018**, *10*, 1412. [CrossRef]
14. Kang, Z.; Yang, J.; Yang, Z.; Cheng, S. A Review of Techniques for 3D Reconstruction of Indoor Environments. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 330. [CrossRef]
15. Yang, P. Research on Image Matching and SFM 3D Reconstruction Method in Mine Environment. Master's Thesis, Xi'an University of Science and Technology, Xi'an, China, 2022.
16. Chen, D.; Wan, L.; Hu, F.; Li, J.; Chen, Y.; Shen, Y.; Peethambaran, J. Semantic-aware room-level indoor modeling from point clouds. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *127*, 103685. [CrossRef]
17. Miao, Y.; Chang, L.; Jun, Q. Neural Radiance Field-Based Light Field Super-Resolution in Angular Domain. *Acta Opt. Sin.* **2023**, *43*, 1411001.
18. Zhang, P.; Hu, G.; Chen, M.; Emam, M. Ambient-NeRF: Light Train Enhancing Neural Radiance Fields in Low-Light Conditions with Ambient-Illumination. *Multimed. Tools Appl.* **2024**, *83*, 80007–80023. [CrossRef]

19. Zimny, D.; Waczyńska, J.; Trzciński, T.; Spurek, P. Points2NeRF: Generating Neural Radiance Fields from 3D Point Cloud. *Pattern Recognit. Lett.* **2024**, *185*, 8–14. [CrossRef]
20. Cao, J.; Yuan, Z.; Mao, T.; Wang, Z.; Li, Z. NeRF-based Polarimetric Multi-view Stereo. *Pattern Recognit.* **2024**, *158*, 111036. [CrossRef]
21. Qin, S.; Xiao, J.; Ge, J. Dip-NeRF: Depth-Based Anti-Aliased Neural Radiance Fields. *Electronics* **2024**, *13*, 1527. [CrossRef]
22. Thomas, M.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *Trans. Graph.* **2022**, *41*, 1–15.
23. Li, Q.; Wang, Z.; Jie, L.; Hu, Y.; Deng, R.; Zhang, H. Dynamic Wind Turbine Blade 3D Model Reconstruction with Event Camera. In Proceedings of the UNIfied Conference of DAMAS, IncoME and TEPEN Conferences (UNIfied 2023), Huddersfield, UK, 29 August–1 September 2023; Ball, A.D., Ouyang, H., Sinha, J.K., Wang, Z., Eds.; Mechanisms and Machine Science, 152. Springer: Cham, Switzerland, 2024.
24. Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *arXiv* **2021**, arXiv:2106.10689.
25. Yariv, L.; Gu, J.; Kasten, Y.; Lipman, Y. Volume Rendering of Neural Implicit Surfaces. In Proceedings of the 2021 Conference on Neural Information Processing Systems (NeurIPS 2021), Online, 6–14 December 2021.
26. Prokopetc, K.; Dupont, R. Towards Dense 3D Reconstruction for Mixed Reality in Healthcare: Classical Multi-View Stereo vs Deep Learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2061–2069.
27. Lu, Y.; Wang, S.; Fan, S.; Lu, J.; Li, P.; Tang, P. Image-based 3D reconstruction for Multi-Scale Civil and Infrastructure Projects: A Review from 2012 to 2022 with New Perspective from Deep Learning Methods. *Adv. Eng. Inform.* **2024**, *59*, 102268. [CrossRef]
28. Li, S.; Li, C.; Zhu, W.; Yu, B.; Zhao, Y.; Wan, C.; You, H.; Shi, H.; Lin, Y. Instant-3D: Instant Neural Radiance Field Training Towards On-Device AR/VR 3D Reconstruction. In Proceedings of the 50th Annual International Syposium on Computer Architecture (ISCA'23), Orlando, FL, USA, 17–21 June 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–13.
29. Croce, V.; Billi, D.; Caroti, G.; Piemonte, A.; De Luca, L.; Véron, P. Comparative Assessment of Neural Radiance Fields and Photogrammetry in Digital Heritage: Impact of Varying Image Conditions on 3D Reconstruction. *Remote Sens.* **2024**, *16*, 301. [CrossRef]
30. Ge, Y.; Guo, B.; Zha, P.; Jiang, S.; Jiang, Z.; Li, D. 3D Reconstruction of Ancient Buildings Using UAV Images and Neural Radiation Field with Depth Supervision. *Remote Sens.* **2024**, *16*, 473. [CrossRef]
31. Wang, X.; Yin, Z.; Zhang, F.; Feng, D.; Wang, Z. MP-NeRF: More Refined Deblurred Neural Radiance Field for 3D Reconstruction of Blurred Images. *Knowl. Based Syst.* **2024**, *290*, 111571. [CrossRef]
32. Wu, T.; Yuan, J.Y.; Zhang, X.L.; Yang, J.; Cao, Y.-P.; Yan, L.-Q.; Gao, L. Recent Advances in 3DGS. In *Computational Visual Media*; Springer: Berlin/Heidelberg, Germany, 2024; Volume 10, pp. 613–642.
33. Jian, G.; Linzhuo, C.; Qiu, S.; Xun, C.; Yao, Y. Advances in Differentiable Rendering Based on Three-Dimensional Gaussian Splatting (Invited). *Laser Optoelectron. Prog.* **2024**, *61*, 1611010.
34. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
35. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
36. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.