3D Reconstruction of Buildings Based on 3D Gaussian Splatting

Zhenglong Cai [1] , Junxing Yang[1] , Tianjiao Wang [1], He Huang[1*],Yue Guo[1]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China - 2108160123001@stu.bucea.edu.cn,yangjunxing@bucea.edu.cn, 2108160122006@stu.bucea.edu.cn,huanghe@bucea.edu.cn,202103010204@stu.bucea.edu.cn

KEYWORDS: Nerf ,point cloud,3D modelling of cities,3D reconstruction,3D gaussian splatting.

ABSTRACT:

In the current era of urban construction, smart city management, and digital twinning, three-dimensional reconstruction of urban buildings is particularly important. Traditional methods have limitations in reconstructing complex geometric scenes, while new methods such as Nerf focus on using implicit MLP to represent the geometric space of the model, but suffer from slow training and rendering speeds. To address this issue, this paper proposes the use of 3D Gaussian scatter points for three-dimensional reconstruction of urban buildings, improving training speed and reconstruction quality through optimized and accelerated rendering algorithms. This method demonstrates high efficiency and editability, providing a new solution for urban building reconstruction.

1. INTRODUCTION

Grids and points are commonly used methods for representing three-dimensional scenes due to their explicit, intuitive, and easily editable characteristics, and they are well-suited for rasterization based on fast GPU/CUDA. Traditional three-dimensional reconstruction methods often employ discrete representations such as triangular meshes or voxel grids, which can faithfully reproduce scenes with complex geometric shapes (Ma & Liu, 2018). However, despite the potential of these techniques to represent complex and high-resolution geometric shapes, they have thus far been limited to simple shapes and low geometric complexity, resulting in overly smooth rendering effects. Particularly when applied to real-world scenes such as urban buildings with low texture, repetitive texture, weak texture, reflection, and highlight areas, these methods fail to represent them with high fidelity.

Furthermore, for the three-dimensional reconstruction of urban buildings, perspective-related and photo-realistic image rendering for visualization become especially important. However, traditional techniques have certain limitations in obtaining perspective-related rendering and synthesizing high-quality new viewpoints. Additionally, the reconstruction and rendering speed of traditional three-dimensional reconstruction techniques for buildings are also limiting factors.

Recent research has focused on Neural Radiance Fields (Nerf) (Wang et al., 2021). This method addresses the synthesis of new viewpoints of objects and three-dimensional reconstruction by encoding objects and scenes in the weights of Multi-Layer Perceptrons (MLP), directly mapping 3D spatial positions to implicit representations of shape. However, achieving high-quality photo-realistic rendering and three-dimensional reconstruction of objects requires training and rendering large MLPs, which may slow down training and rendering speeds. Even recent methods such as InstantNGP (Müller et al., 2022), which use hash grids and occupancy grids to accelerate training and employ smaller MLPs to represent spatial geometric features, still struggle to strike a balance between training speed and model rendering quality.

These methods aim to use implicit MLP representations to display spatial geometry. While large MLPs can store as much spatial features and texture information as possible, they significantly slow down the model's training speed. On the other hand, using smaller MLPs may increase the model's training speed but may result in the loss of many detailed features, thus failing to achieve fine representation of the model. Additionally, this implicit approach makes the model non-editable, whereas three-dimensional reconstruction of urban buildings requires appropriate editing or modification. Clearly, implicit representations and traditional reconstruction methods fail to meet these requirements.

To address the aforementioned issues, we propose the use of three-dimensional Gaussian splats (Kerbl et al., 2023) for the three-dimensional reconstruction of urban buildings. In traditional building three-dimensional reconstruction, RGB point clouds are important features for representing the geometry and color information of the reconstructed object, and point clouds can be edited in real-time to meet the demands of editing building models. Three-dimensional Gaussian splats describe spatial aggregation and texture information of objects through the use of anisotropic and interleaved three-dimensional Gaussian point clouds, distinguishing itself from methods that use implicit MLPs to represent the solid geometry of the model, thus achieving faster model training and better three-dimensional reconstruction results. Firstly, we start with sparse points generated during camera calibration and use three-dimensional Gaussians to represent urban buildings; secondly, we interleave and optimize/density-control three-dimensional Gaussians to achieve accurate representation of buildings; thirdly, we employ a fast visibility-aware rendering algorithm supporting anisotropic splats to accelerate training and achieve real-time rendering. Finally, we develop efficient building model editing methods, allowing us to edit the model as needed (Fang et al., 2023).

We experimentally validate the proposed algorithm using campus datasets. Experimental results on test and evaluation of the datasets of Beijing Architecture University's library, gymnasium, and courtyard at the Daxing Campus demonstrate that the algorithm significantly improves the training speed and surface texture quality of buildings compared to traditional building three-dimensional reconstruction methods and Nerf-based building reconstruction methods.

[*] Corresponding author:**huanghe@bucea.edu.cn**

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-37-2024 | © Author(s) 2024. CC BY 4.0 License.

In summary, the algorithm enhances the speed and effectiveness of urban building reconstruction, making the model editable, and possessing better robustness, scalability, and accuracy than traditional methods.

2. RELATED WORK

Currently, datasets such as KITTI, BDD100K and Cityscapes have complex road scene changes, multiple categories of targets, and different weather conditions, which make them popular datasets among many researchers. For different detection tasks in traffic scenes, there are different requirements for the method of sample collection, the time period of collection, and the collection scene. The quality of dataset collection varies greatly, with differences in the number of samples, resolution, and image target scale, which also directly affect the testing effect and performance of deep learning models. At present, most publicly available datasets are open-sourced datasets from abroad, and there is a lack of publicly available datasets in China. Therefore, this paper uses a self-made dataset for pedestrian and vehicle target detection. The self-made dataset was captured using an HONOR 70 Pro, and it contains two categories: car and person.

2.1 Traditional 3D Reconstruction Methods:

Image-based three-dimensional reconstruction is a key technology in fields such as photogrammetry and computer vision, where the state of the camera at the time of shooting (including camera intrinsic and extrinsic parameters) and the three-dimensional model of the scene (including spatial structure and texture information) are reconstructed from a set of images captured from different viewpoints. It has been widely used in fields such as 3D map surveying, autonomous driving, and smart cities. Generally, this technology consists of the following four main processes:

1. Image orientation: Image orientation techniques recover the camera intrinsic parameters (such as focal length, principal point coordinates, lens distortion coefficients, etc.), extrinsic parameters (such as rotation matrices and translation matrices), and sparse feature points of the photographed scene from a set of input two-dimensional images. In the field of computer vision, this task is usually completed by Structure from Motion (SfM), while in photogrammetry, it is referred to as aerial triangulation, with the sought camera intrinsic and extrinsic parameters also known as interior and exterior orientation elements in photogrammetry.

2. Multi-view dense matching: Multi-view dense matching techniques obtain pixel-level matching relationships based on epipolar constraints between image pairs and pixel similarity, and then use post-processing methods such as graph cuts or filtering to remove erroneous matches. Dense three-dimensional point clouds are then computed through bundle adjustment.

3. Mesh reconstruction and optimization: Mesh reconstruction constructs triangular meshes from dense three-dimensional point clouds to represent the surface of the scene or object. Mesh optimization optimizes the triangular mesh model by adding mesh details and accuracy, such as optimizing the vertex positions of the mesh through multi-view image consistency constraints and continuously iterating updates to make the mesh model closer to reality.

4. Texture mapping: Given the oriented image sequence and the triangular mesh representing the object's surface, texture mapping techniques recover texture maps to describe the appearance of the object's surface. Firstly, the best-view images for each local triangular face are selected for texture mapping, generating preliminary texture maps, and establishing preliminary correspondences between the three-dimensional mesh model and the texture map. Then, color consistency optimization of the textures is performed to avoid large color differences between adjacent texture blocks that may affect visual effects.

Traditional three-dimensional reconstruction algorithms have been developed for many years, with mature theoretical frameworks and commercial software and open-source resources. Their products have been widely applied in various fields of society. With the development of deep learning, three-dimensional reconstruction methods such as MVSNet have emerged, which have improved the quality of three-dimensional reconstruction on specific datasets. However, traditional three-dimensional reconstruction methods have some problems, such as errors in each step from image orientation to texture mapping, which may accumulate over time, leading to increasing deviations from the true values. Additionally, the representation of three-dimensional meshes with textures has limited accuracy and cannot represent details well, such as power lines, trees, moving objects, etc., where problems such as stretching, distortion, and aliasing may occur. Therefore, how to improve the quality of three-dimensional reconstruction, research better representations of three-dimensional scenes, and obtain higher-quality three-dimensional reconstruction results are important issues that need to be addressed urgently.

2.2 Neural Radiance Fields (Nerf):

Early deep learning techniques were used for novel view synthesis. CNNs were used to estimate blending weights or for texture-space solutions. Most of these methods suffer from using geometric structures based on Multi-View Stereo (MVS). Volume representations for novel view synthesis were initially proposed by Soft3D. Subsequent deep learning techniques combining volume ray marching have been proposed, based on continuous differentiable density fields to represent geometric structures. Using volume ray marching for rendering is computationally expensive, as it requires a large number of samples to query the volume. Neural Radiance Fields (NeRFs) introduced importance sampling and positional encoding to improve quality but used a large Multi-Layer Perceptron (MLP) to represent geometry, affecting training and rendering speeds. The success of NeRF has led to a series of subsequent methods, often addressing quality and speed issues through the introduction of regularization strategies; currently, the best technique for novel view synthesis image quality is Mip-NeRF360. Although rendering quality is excellent, training and rendering times remain extremely high; recent methods mainly focus on faster training and/or rendering, mainly by exploiting three design choices:

Using spatial data structures to store (neural) features, interpolation during volume ray marching, and MLP capacity.
Of particular note is InstantNGP, which uses hash grids and occupancy grids to accelerate computation and employs smaller MLPs to represent density and appearance; and Plenoxels, which uses sparse voxel grids to interpolate continuous density fields and can entirely dispense with neural networks. While these two methods provide excellent results, they still cannot effectively represent weak texture areas. Additionally, rendering quality is largely restricted by the choice of structured grid used for acceleration. Our use of unstructured, explicit, GPU-friendly 3D Gaussian functions achieves faster rendering speeds and

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-37-2024 | © Author(s) 2024. CC BY 4.0 License.

better quality.

2.3 3D Gaussian Splatting:

Unlike NeRF, 3D Gaussian Splatting employs an explicit representation and highly parallelized workflow, facilitating more efficient computation and rendering. It combines the advantages of differentiable pipelines and point-based rendering techniques. By using learnable 3D Gaussians ellipsoid functions to represent scenes, it preserves the excellent characteristics of continuous volume radiance fields, which are crucial for high-quality image synthesis, while avoiding the computational overhead associated with rendering in empty space, meeting the demand for faster and more efficient real-time scene reconstruction and rendering.

There have been many related research works, such as: representing complex detail scenes and large-scale scenes from drones requires a large number of 3D Gaussian functions. However, the huge storage space required by Gaussian functions not only hinders their application on devices but also limits rendering speeds. In this context, Lu et al. proposed Scaffold-GS, which maintains comparable rendering quality and speed while being memory-efficient. Additionally, Fan et al. proposed LightGaussian to compress Gaussian functions to improve memory efficiency, making it possible to train and render 3DGS for complex scenes and large-scale scenes from drones in smaller memory conditions and to enhance the compactness and acceleration of rendering complex large scenes. To address aliasing problems that commonly occur during high-quality rendering of complex environmental models, Yan et al. proposed a multiscale approach to alleviate aliasing effects in 3D-GS. By representing scenes at different detail levels and selecting Gaussian functions at appropriate scales, high- and low-frequency signals are effectively encoded, improving rendering quality and increasing rendering speed.

However, most of these reconstruction efforts focus on small object reconstruction, with limited research on urban building scene reconstruction, which is an underexplored area. Therefore, this paper focuses on urban building three-dimensional reconstruction based on 3D Gaussian points rendering technology, seeking applications of 3D Gaussian in urban building scenes and providing a faster and more efficient building reconstruction solution.

3. OVERVIEW

We utilized a set of images of urban buildings obtained from drones and processed them through Structure from Motion (SFM) to generate initial sparse point clouds. These point clouds include the positions (x, y, z) and colors (RGB) information of the scenes within the buildings. We used this sparse initial point cloud information as the foundation for 3D Gaussian reconstruction.

Firstly, we Gaussianized these point clouds in 3D. Specifically, we used the positions of the point clouds as Gaussian means, created an initial covariance matrix and opacity, and represented colors using Spherical Harmonics (SH). We adopted a tile-based rasterizer that allows non-isotropic Gaussian alpha blending, enabling rendering of viewpoint-based building scenes. This method, combined with an adaptive density control module, optimized the building scenes, achieving fast training and rendering.

Using the same dataset, we found that the 3D Gaussian-based building reconstruction method outperformed traditional algorithms in texture details, reconstruction efficiency, and reconstruction quality.

4. METHOD

4.1 PointCloud Gaussianization

We employed a novel approach using 3D Gaussian ellipsoids instead of traditional point clouds to achieve a more refined representation of building scenes. In contrast to the triangular faces and point clouds used by traditional methods, our approach utilizes anisotropic 3D Gaussian point clouds to represent the scene. These point clouds not only contain color and opacity information but also include matrices representing rotation and scaling, allowing these Gaussian point clouds to be distributed in an interleaved manner, better representing building scenes and thus producing more realistic scenes during novel view synthesis.

Our method in this paper utilized initial point clouds obtained from colamp as input, but unlike traditional point clouds and triangular faces, we employed 3D Gaussian functions to characterize complex building scenes. These 3D Gaussian functions are differentiable and can be easily projected onto 2D photographs, enabling fast color blending during rendering. This innovative approach provides a new avenue for the more refined representation and rendering of building scenes.
The Gaussian we use is defined by a full three-dimensional covariance matrix centered at a point (mean), as shown in the following formula:

$$\mathrm{G(x)} = \; e^{-\frac{1}{2}(x)^T \Sigma^{-1}(x)}$$

However, we need to project the three-dimensional Gaussian onto a two-dimensional space for rendering. Given the view transformation matrix W , the covariance matrix in camera coordinates is as follows:

$$\Sigma' = JW\Sigma W^T J^T$$

where J is the Jacobian of the perspective transformation approximation. The covariance matrix of the three-dimensional Gaussian resembles describing the configuration of an ellipsoid. Given the scaling matrix and the rotation matrix, we can find the corresponding:

$$\Sigma = RSS^T R^T$$

To independently optimize these two factors, we store them separately: a three-dimensional vector for scaling and a quaternion to represent rotation. We can convert them into their respective matrices and combine them while ensuring normalization to obtain effective unit quaternions. This representation of anisotropic covariance is suitable for optimization, allowing us to optimize the three-dimensional Gaussian to adapt to different shapes of geometric figures in the captured scene, thus obtaining a relatively compact representation.

4.2 Optimization - Adaptive Density Control

The core of our reconstruction algorithm lies in the optimization step, which involves creating a dense set of three-dimensional Gaussians that accurately represent the scene for free viewpoint synthesis. In addition to position P, covariance Σ, and opacityσ, we also optimize the spherical harmonic (SH) coefficients representing the color for each Gaussian to accurately capture the appearance of the scene as the viewpoint changes. These parameters' optimization is interleaved with the step of controlling Gaussian density to better represent the scene.

4.2.1 Iterative Rendering Optimization:

We optimize based on continuous iterative rendering and compare the generated images with the training views from the captured dataset. Due to the uncertainty of the three-dimensional to two-dimensional projection, geometric placements may occasionally be incorrect. Therefore, our optimization needs to be able to create and destroy or move geometric shapes when they are inaccurately positioned. The quality of the covariance parameters of three-dimensional Gaussians is crucial for compact representation because a few anisotropic large Gaussians can capture large homogeneous areas. We utilize stochastic gradient descent techniques, making full use of standard GPU acceleration frameworks. Notably, the fast rasterization we employ is crucial for optimization efficiency. We use a sigmoid activation function to constrain it within the [0 - 1) range for smooth gradients; for similar reasons, we also use an exponential activation function to compute the scale of covariance.

4.2.2Adaptive Density Control:

We start from the initial sparse point set from SfM and gradually transform it into a denser set of Gaussians through adaptive control of Gaussian quantity and density to better represent the scene. Every 100 iterations, we perform Gaussian densification and remove barely visible Gaussians. We focus on filling missing geometric feature areas and over-covered areas by observing large view space position gradients in both cases. We de-Gaussianize Gaussians with view space position gradients higher than a threshold. For small Gaussians in partially reconstructed areas, we duplicate Gaussians and move them along the position gradient direction to cover newly created geometry. Regions with high variance in large Gaussians need to be partitioned into smaller Gaussians, and we determine the ratio to divide them using experimental coefficients. We initialize their positions based on the original three-dimensional Gaussians as the sampling PDF. We handle them based on the overall system volume and the demand for Gaussian quantity, gradually easing the increase in Gaussian quantity to near-zero after each iteration. Post-optimization, we increase Gaussian density where needed, while allowing our culling method to remove Gaussians below a specified density. This strategy effectively controls the total number of Gaussians without the need for spatial compression, distortion, or projection strategies for distant or large Gaussians.

4.3 Fast Rasterization Module

This paper presents a fast, differentiable rendering method based on 3D Gaussians for synthesizing high-quality new viewpoint images from given camera poses. Our method draws inspiration from the ideas of Neural Radiance Fields (NeRF) and 3D Gaussian Rendering (3D GS), efficiently projecting, sorting, and rendering 3D Gaussians. Firstly, we partition the screen into tiles and discard 3D Gaussians based on the frustum and each tile, retaining only those intersecting the frustum and with a 99% confidence interval. Then, we instantiate each Gaussian based on the number of overlapping tiles and sort them using the GPU. After sorting, we generate a list of Gaussians for each tile through depth sorting and accumulate color and opacity during rasterization until the target saturation is reached. This method succinctly combines projection, sorting, and rasterization to achieve efficient rendering while maximizing parallelism gains. With this approach, we significantly enhance performance during training and rendering while avoiding visible artifacts in converging scenes.

After sorting the Gaussians, we generate a list for each tile by identifying the first and last depth-sorted entries that splat into the given tile. For rasterization, the method launches a thread block for each tile. Each block collaboratively loads Gaussian data packets into shared memory and accumulates color and alpha values for a given pixel by traversing the list from front to back, maximizing parallelism gains in data loading/sharing and processing. When a pixel reaches the threshold for target saturation, the corresponding thread stops. At fixed intervals, the threads in the tile are queried, and processing for the entire tile is terminated when all pixels are saturated (i.e., 0 becomes 1). Saturation of color is the sole stopping criterion during rasterization.

5. EXPERIMENTAL RESULTS AND ANALYSIS

we will discuss some implementation details, showcase results, and evaluate our algorithm against traditional approaches

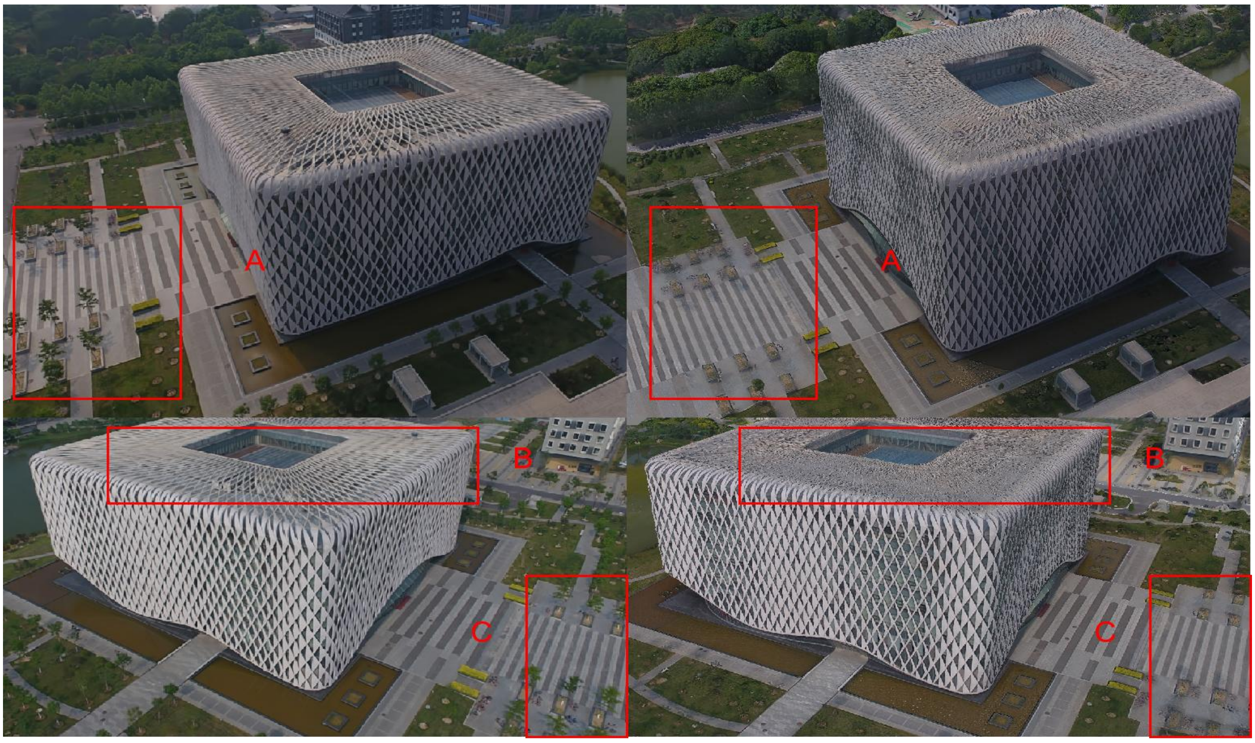

Figure 1
The left image displays the reconstruction of a library scene using the 3D Gaussian method, while the right image showcases the reconstruction result of the traditional method. Through comparison, we observe that in regions A and C, our method outperforms the traditional approach significantly in reconstructing trees around the buildings. Additionally, in region B, our method exhibits superior performance in handling the texture details of repetitive architecture.

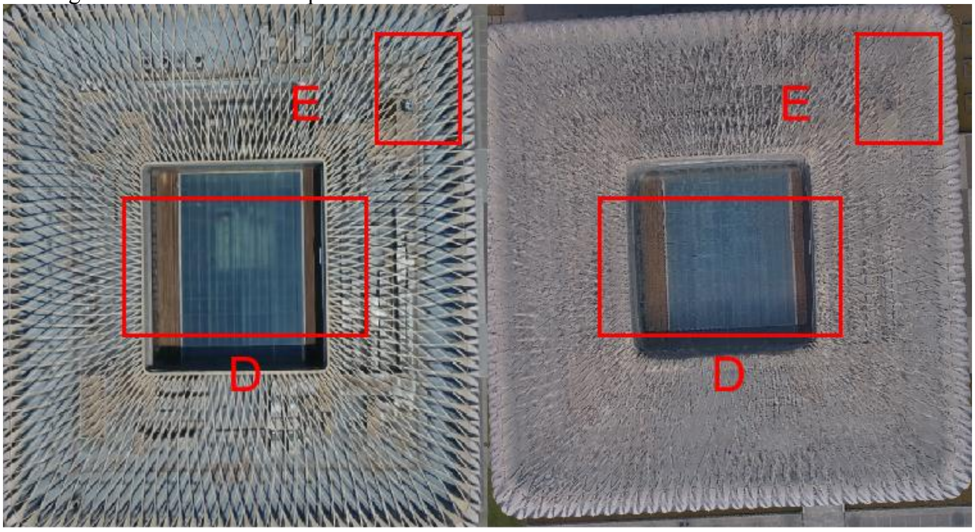

Figure 2
This image highlights the texture details of the steel frame grid and glass material at the top of the building. We can clearly see that our approach better captures the complex texture details of the building, particularly excelling in the reconstruction of special architectural surfaces like glass.

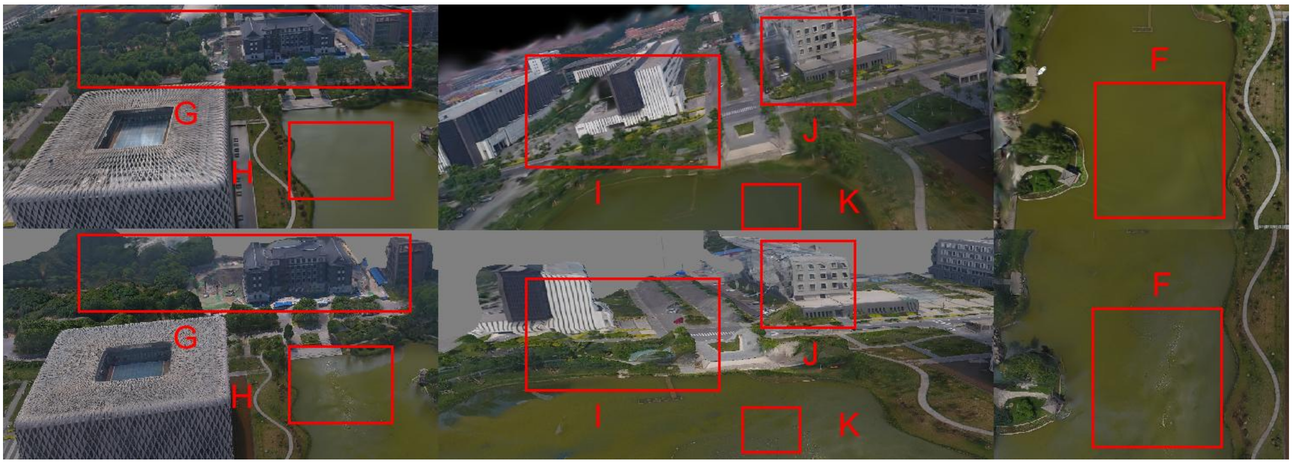

Figure 3
In this comparative illustration, the upper portion showcases the three-dimensional reconstruction of buildings using the 3D Gaussian method, while the lower part presents the results of the traditional approach. It can be observed that, in the traditional method, distant buildings are often neglected, leading to distortions and artifacts in regions G, I, and J. Additionally, inconsistencies in color around the buildings' water bodies are evident in the traditional method, as highlighted in regions H and K of the image.

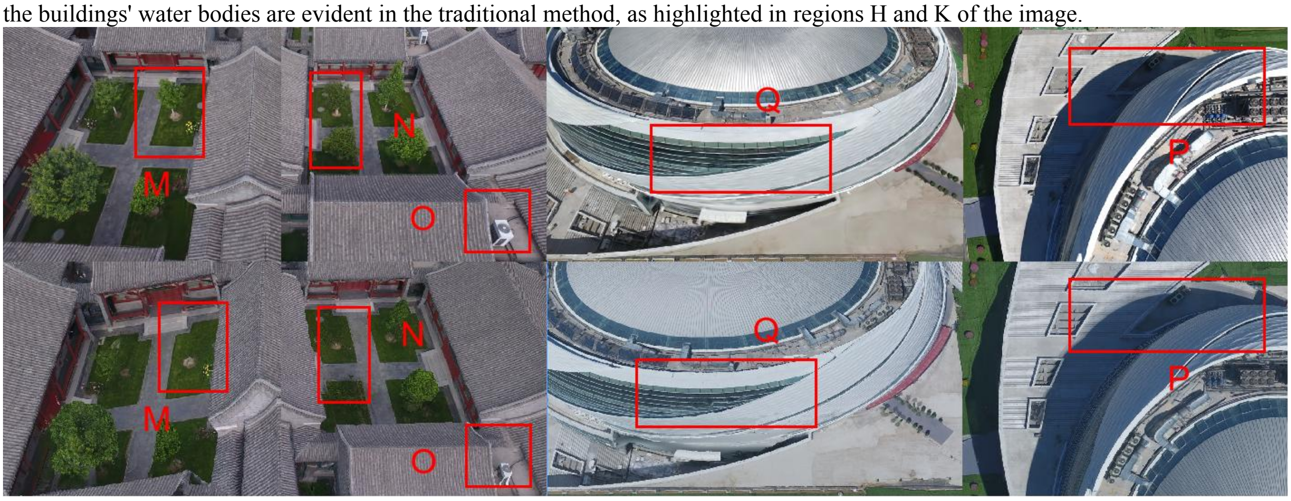

Figure 4
This figure illustrates the comparison of reconstruction results for the quadrangle and library datasets. It is observed that in the traditional method, the reconstruction of trees and air conditioners within the courtyard is less accurate, as evidenced in regions M, N, and O of the image, particularly noticeable distortion in the air conditioner in region O. Additionally, in regions Q and P, distortions are apparent in the glass curtain wall of the gymnasium in the traditional algorithm, and shadows produced by the traditional method in region P exhibit artifacts, which are not present in our approach.

5.1 Implementation：

Considering that the 3D reconstruction of buildings covers multiple scenes and diverse architectural features, we conducted experiments using the campus dataset Cam_datasets obtained from drones. In the experiments, we compared traditional building reconstruction methods with our proposed 3D Gaussian-based reconstruction method for campus buildings. Our data acquisition equipment was the DJI FC330, with a focal length of 3.59357mm and a sensor size of 6.17mm. The equipment used for reconstruction includes a computer equipped with an RTX 3090 graphics card with 24GB of memory.

We implemented this method using Python and the PyTorch framework, utilizing CUDA kernel functions for rasterization. These kernel functions are extensions of previous methods [Kopanas et al. 2021] and employ NVIDIA CUB sorting routines for fast radix sorting [Merrill and Grimshaw 2010]. For interactive visualization, we built an interactive viewer using the open-source SIBR [Bonopera et al. 2020].

We divided the building scenes into three categories: library, ancient quadrangle, and gymnasium. These scenes not only contain individual buildings but also include surrounding environmental information such as trees, water bodies, etc. In these scenes, we conducted comparative experiments using both traditional methods and the method proposed in this paper. Throughout the experiments, we maintained consistent input data and equipment configurations to ensure a fair comparison between our proposed method and traditional methods in terms of 3D building reconstruction.

5.2 Result and Evaluation:

In terms of reconstruction results, we observed that the method proposed in this paper outperforms traditional methods in overall building reconstruction. Specifically, our method performs better in complex texture details, reflective surfaces, and glass materials. These advantages are demonstrated in Figures 1 and 2. Additionally, in Figure 3,

Method	Libaray	Yard	Gym
Ours	28min	25min	35min
Tradition	42min	47min	51mim

we found that traditional methods exhibit distortion and stretching in the reconstruction of objects around buildings, whereas our method avoids these issues and more faithfully reproduces the surrounding environment such as trees. In special reconstruction areas such as water bodies, the results in Figure 3 further demonstrate the consistent advantage of our method over traditional methods.

6. CONCLUSION

This paper introduces an innovative approach to architectural reconstruction based on 3D Gaussian splatting, utilizing anisotropic 3D Gaussian ellipsoids to represent scenes. This approach offers possibilities for high-quality rendering and training, surpassing traditional methods as well as the point cloud and MLP used in Nerf. We employ a fast and differentiable rasterization module, accelerating the entire training and rendering process with GPU. Notably, our method outperforms traditional algorithms in both reconstruction speed and model rendering quality, especially in aspects such as texture details, water bodies, and specular materials. By showcasing Gaussian point cloud representations of scenes, we enable scene editing, providing new insights for the three-dimensional reconstruction of urban buildings.

This Gaussian-based method accurately captures the complexity and details of architectural scenes and effectively handles rendering and synthesis from various perspectives. Additionally, our method offers greater editability, allowing users to flexibly modify and optimize scenes. Therefore, we believe this approach will lead to significant advancements in the field of three-dimensional reconstruction of urban buildings and provide new perspectives and methods for related research.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant Numbers 42201483) and the China Postdoctoral Science Foundation (Grant Numbers 2022M710332).

REFERENCES

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In Proceedings of the 9th European Conference on Computer Vision - Volume Part I (ECCV '06) (pp. 404-417).

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05) (Vol. 1, pp. 886-893).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 248-255).

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and sesegmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In European conference on computer vision (pp. 740-755).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European conference on computer vision (pp. 21-37).

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91-110.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2019). Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 8430-8439).

Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781-10790).