Fast 3D Environment Reconstruction from 360 Video

WANG WENFAN[1] r12922161@ntu.edu.tw, LEE TINGYING[1] r12725054@ntu.edu.tw, and LIN SHINHUNG[1] r12922117@ntu.edu.tw

National Taiwan University

Abstract. In this study, we explore the utilization of 360-degree cameras for fast 3D environment reconstruction, focusing on the integration of NeRF (Neural Radiance Fields) and Gaussian splatting techniques. We detail our approach to capturing and processing 360-degree videos, including challenges and solutions related to feature matching and the generation of point clouds. The application of our technique in various scenarios, including indoor and outdoor environments, demonstrates its versatility and effectiveness. This paper provides insights into the potential of advanced 3D reconstruction methods in rapidly generating high-fidelity models from 360-degree video data.

Keywords: 360 camera · NeRF · Gaussian splatting

1 INTRODUCTION

The modeling of 3D scenes finds widespread applications in entertainment, education, and various aspects of daily life.

However, the conventional manual methods of creating these models are labor-intensive. Hence, our goal is to introduce an automated solution. We plan to design an application that leverages a 360-degree video to seamlessly construct a 3D scene.

To effectively reconstruct a 3D environment, NeRF relies on a substantial collection of photos captured from various perspectives throughout the entire scene. Nevertheless, 360-degree videos offer a convenient means of swiftly obtaining such images from real-world surroundings.

The advantages of 3D scene reconstruction can be harnessed across a wide range of applications, particularly when time is of the essence. Consider Google Maps' Street View, which facilitates outdoor navigation for users. Unfortunately, indoor environment information is often lacking. NeRF, however, offers a straightforward solution by enabling the effortless reconstruction of indoor spaces. This, in turn, empowers individuals to navigate within buildings more easily.

Furthermore, this capability holds significant importance for individuals with visual impairments. With access to an indoor map, an AI assistant can swiftly chart optimal routes for these individuals, assisting them in reaching their destinations more efficiently.

2 RELATED WORK

2.1 3D Gaussian splatting

Since the release of NeRF, the field of view synthesis research has experienced a remarkable surge in activity. Nevertheless, the extensive training and prediction times associated with NeRF render it impractical for real-time applications. In addressing this issue, 3D Gaussian splatting, as discussed in [1], offers an alternative solution that enables real-time scene rendering with significantly lower computational demands.

3 METHOD

We captured 360-degree videos using an Insta360 camera on the campus of National Taiwan University. Our initial goal was to create an indoor navigator, but photographing inside the NTU main library was prohibited. As a result, we opted to film in the corridors of an indoor building. To avoid capturing people in the video, we chose to film in the morning, a time when there were almost no pedestrians.

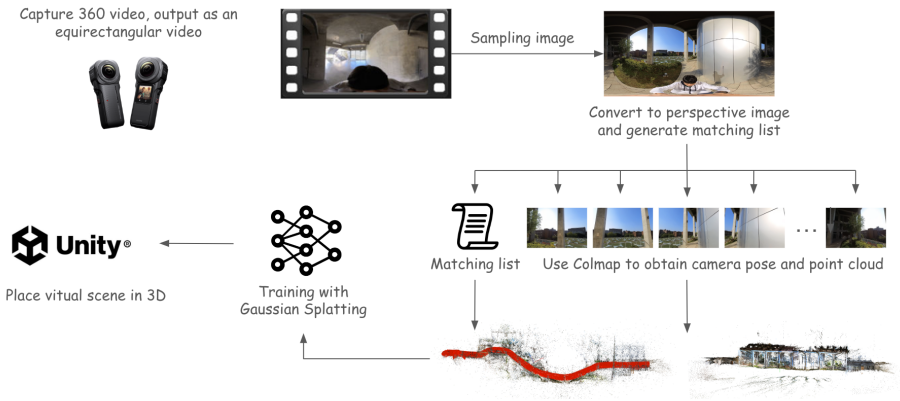

Fig. 1. Initially, when a panorama video is fed as input, it splits into frames of equirectangular images, which are subsequently converted into perspective images. In the second stage, we utilize the COLMAP tool to derive camera poses and develop a point cloud. The third stage involves feeding the obtained point cloud into a Gaussian splatting process to create a 3D scene. Finally, this generated 3D scene is integrated into the Unity environment for the final output.

3.1 Data preprocessing

After filming, we use Insta360 Studio to output the 360-degree videos as panorama videos. Each frame in these panorama videos is an equirectangular image. Unlike cylindrical views, spherical panoramas incorporate a 180° vertical viewing angle and a 360° horizontal viewing angle.

First, we need to perform equirectangular projection to transform these equirectangular images into perspective images. The most common approach for this transformation is the cubic format, which uses six cube faces to cover the entire sphere surrounding the viewer. These maps are often created by capturing the scene with six cameras, each having a 90-degree field of view (FoV), providing left, front, right, back, top, and bottom textures. The six images are typically arranged as an unfolded cube, which is why it is called the cubic format.

To avoid distortion, we do not use the top and bottom images, but only the images from the left, front, right, and back directions. Then, since we need to perform feature matching in subsequent steps, it is essential to ensure overlap between images. Therefore, we cannot simply divide the panorama into four images; instead, more images are required to guarantee sufficient overlap. We chose a frame rate of 1 fps, slicing each frame into six images from different angles. Thus, every second, we obtain six images.

Fig. 2. Result of convert equirectangular image to multiple perspective images

3.2 Structure from motion

We use COLMAP for structure from Motion, with steps including feature extraction, feature matching, bundle adjustment, and reconstruction. In the matching phase, COLMAP offers several methods: exhaustive and sequence matching. Exhaustive matching involves pairwise comparison of all input images, resulting in n^2 pairings, which is very time-consuming. Sequence matching, on the other hand, treats the input images as a sequence from a video, where each image is matched with its immediate predecessors and successors. However, this approach only yields a point cloud from a single perspective. Our input data, however, involves both temporal and spatial matching. Therefore, we defined a custom matching list that allows each image to match not only with its adjacent images in the same frame but also with images of the same angle from preceding and following frames. This significantly enhances accuracy and efficiency.

Fig. 3. colmap

3.3 NeRF

We have chosen to use NeRF (Neural Radiance Fields) as the first method. NeRF is a three-dimensional reconstruction method based on neural networks, designed to generate realistic 3D scenes from two-dimensional images. Its objective is to train a neural network model to predict the color and density of any point in three-dimensional space. The core idea of this method is to use a neural network to represent the surfaces in a 3D scene, rather than traditional point clouds or grid representations.

Our implementation involves first using COLMAP to obtain camera poses and point clouds.Subsequently, we utilize Instant-NGP for training the model to reconstruct the 3D scene.

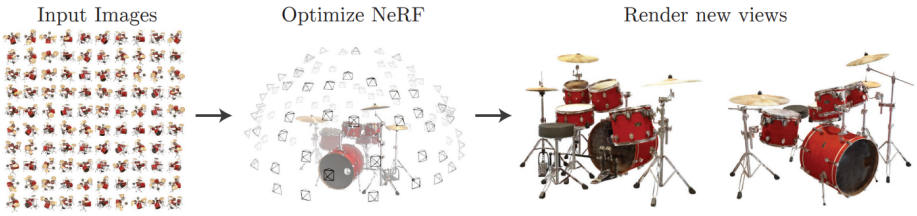

Fig. 4. NeRF

3.4 Gaussian Splatting

We have chosen to employ 3D Gaussian splatting, as detailed in the work by Kerbl et al. [cite: kerbl20233d], as our ultimate solution.

Gaussian Splatting serves as a rasterization technique for the real-time reconstruction and rendering of 3D scenes using images captured from various viewpoints. This approach allows us to generate high-quality, photorealistic scenes while maintaining fast, real-time rasterization capabilities.

Our implementation begins by acquiring camera poses and point clouds through the COLMAP framework. Subsequently, each individual point within

the point cloud is transformed into a Gaussian, enabling the rasterization process. These Gaussians are characterized by four key parameters: Position, Covariance, Color, and Density, which are fine-tuned during the training phase. To obtain the Gaussian representations from images, we employ differentiable Gaussian rasterization. The loss is then computed by comparing the generated rasterized image with the ground truth.

During the prediction phase, when provided with the camera pose as input, each Gaussian must be projected into 2D from the camera's perspective. These Gaussians are then sorted by depth, and for each pixel, iterate over the Gaussians in a front-to-back order, blending them together to produce the final output.

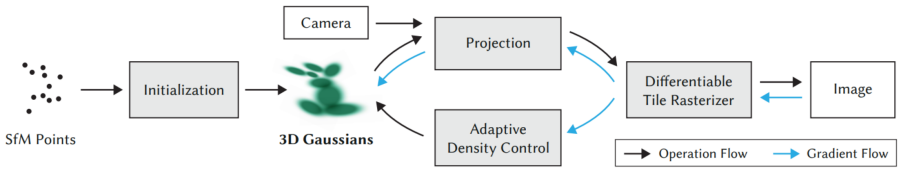

Fig. 5. gaussian splatting

3.5 Reconstruction in game engine

Using Gaussian splatting, we generated a point cloud file post-training that encompasses the parameter details of all the Gaussians. Leveraging the solution developed by Aras Pranckevičius[3], we import these Gaussian into Unity. From there, we have the capability to construct a convincing 3D scene directly from a 360-degree video within the game engine. Moreover, the program provides the flexibility to fine-tune parameters such as opacity strength to achieve the desired visual effects.

4 RESULTS

We filmed the corridor of Academic Affairs Office and the Main Library, the indoor area of the first floor of CS department, Der Tian Hall, and the landscaped third floor of the College of Social Sciences Building at National Taiwan University. Our results will be presented in the form of images and videos.

4.1 Instant-NeRF

We use the built-in renderer provided by Instant-NGP to render our predictions. The following image 6 is one of the results.

Fig. 6. Result of instant NeRF

4.2 Gaussian Splatting

We render the prediction using the SIBR Viewers provided by the 3D Gaussian Splatting team and Unity. Below 7 and 8 are multiple examples of our rendering results and comparison.

Fig. 7. Result of Gaussian splatting

We also provide the link to our render result videos for examination: Link to our videos.

Fig. 8. Comparison of points and splatts

5 DISCUSSION

5.1 Structure from motion

Since the input for Gaussian splatting is a point cloud, structure from motion is crucial. Therefore, improving feature matching is necessary. Our initial approach involved using a FOV of 90°, with 45° per image for a total of 8 images, for 3D reconstruction. However, the results were not ideal. For instance, there were holes in some areas. We noticed that many of these holes were located in shadowed areas. Tracing back to the corresponding input images, we realized that this issue was largely due to simple textures combined with shadows. Simple textures make feature matching challenging, leading to a situation where the matched points are more likely from shadows rather than textures. Since shadows appear differently when viewed from various positions and angles, this discrepancy caused inaccuracies in the matching results. Therefore, we ultimately decided to increase the FOV. This change was made to enhance the content of each image, thereby reducing the instances where a single texture was the only subject captured in a picture.

5.2 Training

Throughout the training process, we observed that varying the learning rates had a pronounced effect on our outcome, particularly in the context of scenes with varying scales.

Following numerous experiments, we discerned that a lower learning rate for positional adjustments is better suited for larger scenes. This finding aligns with expectations, as larger scenes inherently require more precise positional information to yield satisfactory results.

Additionally, the incorporation of a matching list in our training process yielded substantial improvements. Utilizing this matching list consistently resulted in higher PSNR metrics, indicative of enhanced training outcomes.

5.3 Applicable Scenarios

We have observed that distant views yield better results compared to close-up views. For instance, the pillars in the corridor may not appear very clear, but the scenery outside the corridor is quite distinct. This could be due to the fact that feature matching is easier for distant views, resulting in more available information. In contrast, close-up views have relatively less information, leading to a sparser generation of points in the point cloud. Additionally, shadows and reflections can severely impact the quality of the results. However, it's worth noting that while reflections from windows do affect the outcome, the scenery visible through the windows remains clear.

We have found that the best conditions occur in wide indoor or outdoor scenes, along with fixed lighting situations. This helps to avoid sudden overexposure that can affect image quality and also reduces the impact of shadows and reflections.

6 LIMITATIONS AND FUTURE WORK

6.1 Limitations

When capturing 360-degree videos for 3D reconstruction or similar purposes, the presence of people can introduce unwanted movement and variability. This can complicate the process of feature matching and 3D modeling, as moving objects (like people) can disrupt the consistency of the scene, making it difficult to stitch images together accurately. Consistent lighting is crucial for maintaining uniformity in the captured images. Fluctuations in lighting can lead to variations in shadows and highlights, affecting the reliability of feature detection and matching. Consistent lighting helps in maintaining the integrity of the textures and details in the scene, which is essential for accurate 3D reconstruction and rendering. Therefore, it is necessary to avoid people and ensure a consistent lighting situation when filming.

Since we are capturing 360-degree videos, we are able to obtain both temporal and spatial correspondence information within the same video. However, this type of self-defined matching list is only applicable to a single video. If we want to film multiple videos of the same scene from different heights and angles, we would have to resort to exhaustive matching.

6.2 Future work

Data quality enhancement In the future, we aspire to enhance video quality by incorporating automatic object removal for people, as well as eliminating shadows and reflections. These improvements can be achieved through the utilization of cutting-edge deep learning techniques.

Better Matching Method The outcome of our 3D reconstruction process is greatly reliant on the point cloud generated through structure-from-motion. In our current approach, we employ COLMAP, which utilizes SIFT for feature matching. In future research, employing deep learning-based methods may yield superior results.

7 CONCLUSION

In conclusion, our project successfully demonstrates the potential of 360-degree cameras in conjunction with NeRF and Gaussian splatting for rapid and efficient 3D environment reconstruction. We have addressed several challenges inherent to this approach, including feature matching and the impact of lighting conditions. The research highlights the importance of accurate data capture and processing techniques in achieving high-quality 3D models. While there are limitations in terms of shadows and reflections, our findings open avenues for further improvement and optimization. Future work may focus on enhancing the algorithm's robustness to diverse lighting conditions and incorporating advanced object removal techniques to refine the reconstruction process.

References

1. Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." ACM Transactions on Graphics 42, no. 4 (2023)
2. Müller, Thomas, Alex Evans, Christoph Schied, Marco Foco, András Bódis-Szomorú, Isaac Deutsch, Michael Shelley, and Alexander Keller. "Instant neural radiance fields." In ACM SIGGRAPH 2022 Real-Time Live!, pp. 1-2. 2022.
3. [aras-p/UnityGaussianSplatting: Toy Gaussian Splatting visualization in Unity (github.com)](https://github.com/aras-p/UnityGaussianSplatting)