Project Proposal
3D Reconstruction for Vehicle

Chau Vinh Dat, 22KHMT, *22127056*
Ngu Kiet Hung, 22TGMT, *22127134*

***Abstract*—3D reconstructions by synthesizing novel views of complex scenes has achieved many success in the coming years. From Neural Radiance Field [6], a scene can be represented using a fully-connected deep network, and produce appearance based on a continuous 5D coordinate (spatial location (x, y, z) and view direction (θ, ϕ)). From then numerous variant which optimize the performance in both quality and execution time of NeRF [1, 12, 14] have pushed forward the boundary on both quality and performance, but there is always a tradeoff between the two criterion. 3D Gaussian Splatting [5] introduce three key elements that allow us to achieve state-of-the-art visual quality while maintaining competitive training times and importantly allow high-quality real-time ($\geq$ 30 fps) novel-view synthesis at 1080p resolution. This project aims to re-implement the method described in [5], and experiments the method on vehicles scene reconstruction. An extension of the project is to accurately and densely reconstruct the 3D mesh data of the vehicle using a combination of 3D Gaussian Splatting and SuGaR [4].**

***Index Terms*—3D Reconstruction, Vehicles, 3D Gaussian Splatting, Neural Radiance Field, Spherical Harmonics.**

I. Introduction

A. Motivation

3D VEHICLE reconstruction is an essential component of several cutting-edge technologies, ranging from autonomous driving and robotics to virtual reality, gaming, and digital twins. As the demand for precise and efficient 3D models increases, the methods used for reconstructing these models must evolve to meet the growing challenges of performance, scalability, and visual fidelity [2, 3, 7].

B. Objective and Scope

While 3D reconstruction techniques, such as Structure from Motion (SfM) and Multi-View Stereo (MVS) have made significant strides in capturing complex 3D scenes, these methods often struggle with the specific challenges posed by reconstructing vehicles. Furthermore, traditional methods may require extensive computational resources, limiting their ability to perform real-time reconstruction for on-line interaction.

In this project, we aim to:

- Find a representation of vehicle in 3D space, which allow rendering of view (Multi-View Stereo task).
- The approach should be efficient in terms of memory and rendering speed.
- The reconstructed 3D object should be dense, therefore, sparse 3D reconstruction steps (e.g Struction from Motion) for feature extraction can be skipped.

II. Problem Statement

3D reconstruction from our point of view is the Multi-View Stereo task: reconstruct a 3D representation of a scene from multiple images of different views, given the known camera poses. The output should be rendered images of desired views. As we discover the methodologies of different techniques, the rendered scene may be

III. Related work

A. Neural Radiance Fields

Neural Radiance Fields (NeRF) [6] utilize the idea of volumetric rendering, that is to render a pixel of an image of a 3D scene, we render the color of the corresponding ray passing through the scene. The color C is given by:

$$C(r) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i \delta_i))c_i \tag{1}$$

with δ_i is a interval between some sampled t_i and t_{i+1} along the ray $r(t) = o + td$, σ_i is the volume density at sample t_i - which can also be interpreted as the probability of the ray terminating at an infinitesimal particle at location t_i, and $T_i = exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ is the accumulated transmittance up to t_i - the probability of the ray traveling to t_i without hitting any particle.

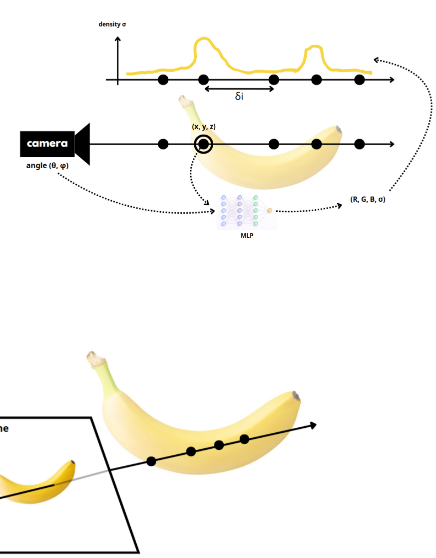

Fig. 1: Volumetric rendering with NeRF.

The NeRF model is a multilayer perceptron or MLP which takes the 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction (θ, ϕ) as input and outputs the color $c = (r, g, b)$ and volume density σ. By using this appoarch, a representation of the 3D scene can be stored in the form of the weights in the MLP, which can be used to render the color of pixels for required view.

NeRF also leverge other method to improve the quality of rendered image such as positional encoding and hierarchical sampling, however, the main idea of the model is described as above.

B. NeRF variances

An obvious downside of NeRF is that NeRF models are very slow, since for every pixels, the neural network has to run various datapoint for a single ray. The model Plenoxel [14] remove the neural network and use a voxel grid with spherical hamornics to interpolate the radiance field. InstantNGP [7] leverge hashing the parameter encoding to improve the training time and rendering time of the model.

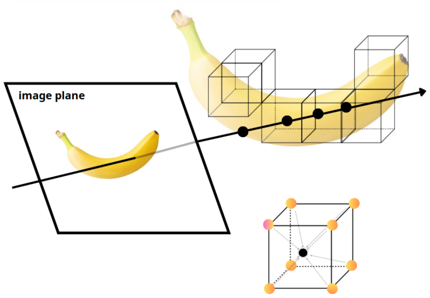

Fig. 2: Spherical Hamornic interpolation with Plenoxels

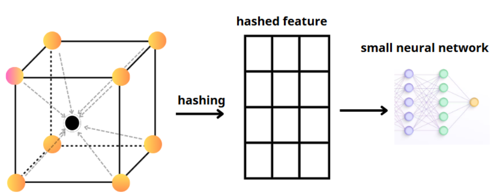

Fig. 3: InstantNGP hashing encoding feature

Mip-NeRF improve the quality of fine detail by efficiently rendering anti-aliased conical frustums instead of rays.

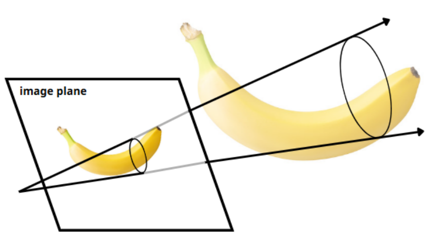

Fig. 4: Mip-NeRF methodology.

There are more variances of NeRF, however, the main ideas of the models are as presented.

C. Surface reconstruction

A problem of NeRF models is that they only allow rendering out the 3D scene onto the viewed 2D plane, which means there is no real surface of the 3D object. If the 3D surface is reconstructable, various application such as meshing or object collision can be performed.

One possible solution to reconstruct 3D surfaces is to use the volume density σ above a threshold as the surface. However, this approach results in noisy 3D surfaces, since predicted the volume density may fluctuate around the true surface as different levels.

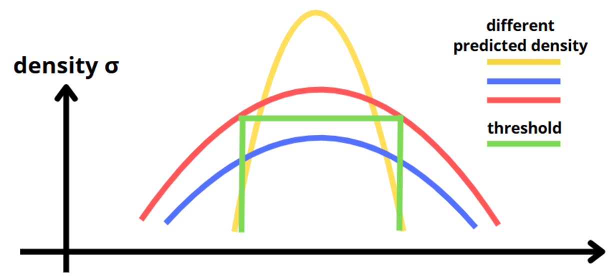

Fig. 5: Problem with surface rendering with volume density threshold.

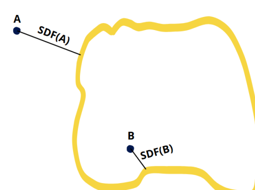

Fig. 6: Signed distance function.

The model such as NeuS [10] or VolSDF [12] solve the problem by using SDF - signed distance function, instead of volume density for surface representation. The function SDF takes a position and return the shortest distance from it to the surface of the desired object, the signed value describe the point inside or outside of the surface. The advantage of this method is that the set of zero levels SDF (i.e $\{x \in \mathbb{R}^3 | SDF(\mathbf{x}) = 0\}$) defines the surface geometrically well.

D. 3D Gaussian Splatting

3D Gaussian Splatting took another approach into the task of 3D reconstruction, by fitting a set of 3D Gaussian to re-create the scene. 3D Gaussians are empirically chosen as the primitive for this model, since they are differentiable and can be easily be projected into 2D plane and rendered via α-blending and rasterization.

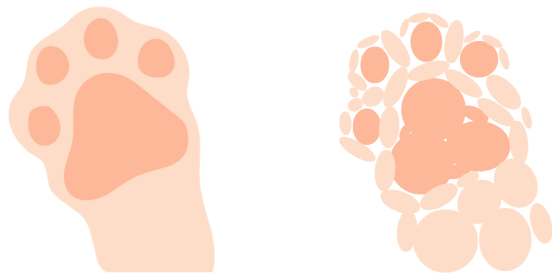

Fig. 7: Gaussian Splatting illustration.

The illustration above describe the idea of the Gaussian splatting (the ovals representing Gaussians). The quality of representation is enhanced for fine detail as the the Gaussian becoming microscopic.

A Gaussian is defined with a full 3D covariance matrix Σ as:

$$G(x) = exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{2}$$

with the mean μ as the center point, Σ denoting the shape of the Gaussian:

$$\Sigma = RSS^TR^T \tag{3}$$

E. Generative Models

3D reconstruction from 2D images are difficult in case of lacking images multiple view. The methods described above share the same input as SfM (Structure from Motion), meaning the structures are recreated from different view of static objects (”motion” indicating the viewd camera moving around the scene). Although our goal is not reconstructing object from these scenarios, it is worth mentioning that generative models using diffusion can be used filling in the lack of information. The idea of generative 3D reconstruction can be found in model such as DreamGaussian [9], DreamCraft3D [8], or more particularly DreamCar [3].

IV. Proposed Methodology

Due to the computation complexity of many proposed works, as well as considering the performance of state-of-the-art methods, it is necessary to ensure a balance between visual quality and competitive training time, as well as real-time rendering performance.

Our goal is to allow real-time rendering for scenes captured with multiple photos, and create the representations with optimization times as fast as the most efficient previous methods for typical real scenes.

In this work, we aim to use 3D Gaussian Splatting for reconstructing 3D vehicles. The motivation behind this choice is:

- As each time the view changes, NeRF-like models suffer from speed performance issue since the entire view has to be rerendered. 3DGS offers better realtime performance in terms of speed and quality.
- Since the Gaussians are explicitly created to be rendered, reconstructed objects can be transformed in terms of position or rotation, etc.

V. Dataset

A. Overview of the chosen dataset

Similar to other SfM-based methods, 3D Gaussian Splatting requires multiple view of the scene to perform the reconstruction. We choose the 3DRealCar [2] dataset for it properties:

- High volume of about 2500 cars scanned.
- Each consists an average of 200 high resolution 360° views.
- Most importantly, it capture the most important car-specific property: made of glossy material (metal or glass).
- The dataset also contains cars in dark environment to test the model in awful conditions.

B. Discussion on the vehicle data and potential solution

As described, the materials of car surfaces are generally glossy, which means it would produce plenty of specular highlights if cars are exposed to the sun or strong light. This may cause trouble since SfM models often not taking reflected lighting into account.

One approach for this problem might include recreating the surface of the vehicle to perform lighting methods such as shader. However, there remain difficulties detecting the reflective surface and detecting the lighting source in 3D reconstruction. Another possible method can be utilizing generative models for generating reflective texture on the geometry of the reconstructed object, but this approach may require further experiment.

For recreating geometry with 3D Gaussian, recent models such as SuGaR [4] or GauS [13] have enabled mesh recreation from 3D Gaussian splats. However, the problem of reflective surface in reconstruction remains unsolved.

VI. Evaluation Method

A. Structural Similarity Index Measure (SSIM)

SSIM [11] is a commonly used method to measure the similarity of 2 images. For two images x and y, for each (i, j) pixel, SSIM is given by:

$$SSIM(i,j) = [l(i,j)]^\alpha[c(x,y)]^\beta[s(x,y)]^\gamma \tag{4}$$

where the *luminace* l, *contrast* c, *structure* s are:

$$l(i,j) = \frac{2\mu_x(i,j)\mu_y(i,j)+c_1}{\mu_x(i,j)^2+\mu_y(i,j)^2+c_1}$$

$$c(i,j) = \frac{2\sigma_x(i,j)\sigma_y(i,j)+c_2}{\sigma_x(i,j)^2+\sigma_y(i,j)^2+c_2}$$

$$s(i,j) = \frac{\sigma_{xy}(i,j)+c_3}{\sigma_x(i,j)\sigma_y(i,j)+c_3}$$

where the mean μ and variance/covariance σ are functions of the window around the pixel, computed by 11×11 Gaussian-weighted window with $\sigma = 1.5$. The original paper of SSIM propose that $c_1 = (k_1L)^2$, $c2 = (k_2L)^2$ and $c3 = c_2/2$ where $L = 255$ for 8-bit images, $k_1 = 0.01$ and $k_2 = 0.03$.

Furthermore, we can simplify the equation by setting $\alpha = \beta = \gamma = 1$, which results in:

$$SSIM(i,j) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \tag{5}$$

B. Peak Signal-to-Noise Ratio (PSNR)

PSNR is a normalized measure for Mean Square Error:

$$PSNR = 10\log_{10}(\frac{R^2}{MSE}) \tag{6}$$

where $R = 255$ is the maximum possible value of the 8-bit pixel.

The larger PSNR is, the more accurate the rendered image as the MSE is smaller. PSNR is used for comparing rendered images in different settings, e.g. $R = 2^{16}$ for 16-bit images.

VII. Project timeline

Based on initial survey of the topics, as well as the progression of our individual research, a timeline of joint contribution to the project is laid out through the table below:

TABLE I: Project timeline

Week (with respect to the course timeline)	Tasks
Week 8-9	Re-producing the results of 3D Gaussian Splatting original publication [5].
Week 9-10	Implementing SuGaR [4] for mesh surface reconstruction.
Week 11	Results data gathering, analysis and report finalization.

VIII. Conclusion

In this report, we explored various approaches to 3D reconstruction, with a particular focus on methods that can be utilized for vehicle-specific data. For our specific case, we proposed 3D Gaussian Splatting (3DGS) as a promising approach for vehicle reconstruction on the 3DRealCar dataset. For future advantages, additional advancements in handling reflective surfaces and improving generative models could further enhance the quality and applicability of the technique.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844. IEEE. ISBN 978-1-6654-2812-5. doi: 10.1109/ICCV48922.2021.00580. URL https://ieeexplore.ieee.org/document/9710056/.

[2] Xiaobiao Du, Haiyang Sun, Shuyun Wang, Zhuojie Wu, Hongwei Sheng, Jiaying Ying, Ming Lu, Tianqing Zhu, Kun Zhan, and Xin Yu. 3drealcar: An in-the-wild RGB-d car dataset with 360-degree views. URL http://arxiv.org/abs/2406.04875.

[3] Xiaobiao Du, Haiyang Sun, Ming Lu, Tianqing Zhu, and Xin Yu. DreamCar: Leveraging Car-specific Prior for in-the-wild 3D Car Reconstruction, July 2024. URL http://arxiv.org/abs/2407.16988. arXiv:2407.16988 [cs].

[4] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. URL http://arxiv.org/abs/2311.12775.

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering, August 2023. URL http://arxiv.org/abs/2308.04079. arXiv:2308.04079 [cs].

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, August 2020. URL http://arxiv.org/abs/2003.08934. arXiv:2003.08934 [cs].

[7] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022. ISSN 0730-0301, 1557-7368. doi: 10.1145/3528223.3530127. URL http://arxiv.org/abs/2201.05989. arXiv:2201.05989 [cs].

[8] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior, October 2023. URL http://arxiv.org/abs/2310.16818. arXiv:2310.16818 [cs].

[9] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation, March 2024. URL http://arxiv.org/abs/2309.16653. arXiv:2309.16653 [cs].

[10] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction, February 2023. URL http://arxiv.org/abs/2106.10689. arXiv:2106.10689 [cs].

[11] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612, 05 2004. doi: 10.1109/TIP.2003.819861.

[12] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume Rendering of Neural Implicit Surfaces, December 2021. URL http://arxiv.org/abs/2106.12052. arXiv:2106.12052 [cs].

[13] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuantao Chen, Yihao Zhi, and Xiaoguang Han. Gaustudio: A modular framework for 3d gaussian splatting and beyond. *arXiv preprint arXiv:2403.19632*, 2024.

[14] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks, December 2021. URL http://arxiv.org/abs/2112.05131. arXiv:2112.05131 [cs].