Specular and Spectacular: Accurate 3D Reconstruction of Automotive Surfaces with Gaussian Splatting

Bohan Jeff Li
Stanford University

bohan1@stanford.edu

Abstract

High fidelity 3D reconstruction of vehicles from limited photographic inputs presents significant challenges due to their reflective and complex surfaces. This task is crucial in applications ranging from gaming and virtual marketplaces to autonomous driving technologies, attracting continuous interests within the computer vision field. Traditional methods often struggle with specular reflections and surface texture variations, leading to imprecise models. This study explores the efficacy of Gaussian Splatting, a relatively novel point-based rendering technique, in overcoming these challenges. Gaussian Splatting utilizes ellipsoidal Gaussian distributions to represent radiance functions in 3D space, ensuring smoother and more continuous surface representations. This method effectively models a wide range of surface illumination phenomena, accurately capturing both diffuse and specular reflections while reducing misinterpretations of them as geometry. We applied Gaussian Splatting to reconstruct 3D models of vehicles using high-resolution video captures from consumer-grade smartphones. Our results demonstrate that Gaussian Splatting not only enhances the visual fidelity of the models but also maintains computational efficiency, making it feasible for at-home applications. The findings suggest that Gaussian Splatting holds significant promise as a new paradigm in 3D reconstruction, indicating potential for widespread adoption across various industries, paving the way for future development in the field. Our code and collected dataset are available at https://github.com/voidJeff/AutomotiveGaussianSplatting

1. Introduction

Accurately reconstructing 3D models of cars from a limited set of photographic views presents an interesting blend of challenges and opportunities for computer vision. This task is not only technically demanding due to the complex, reflective surfaces of automobiles but also highly relevant to numerous applications ranging from gaming to virtual marketplace to autonomous driving technologies. We want to address this challenge from both an automotive industrial and computational engineering perspective, bridging creative passions with beneficial societal implications. Furthermore, the potential applications of 3D reconstruction in AR/VR environments for business, entertainment, and education highlight the broad impact and commercial viability of refining these technologies.

This project aims to explore the broad field of photogrammetry by incorporating specific innovative techniques to handle the specular reflections typical of car surfaces, which traditional methods often struggle to manage. The selection of cars as the focus of this project provides opportunities for additional algorithmic adaptations in preprocessing and hyper-parameter tuning, as cars possess inherent properties such as bilateral symmetry and consistent rim sizes across models, which might introduce additional constraints that can significantly enhance reconstruction accuracy. These features enable a unique chance to refine existing methods and probe the boundaries of what can be achieved with current computer vision technology.

2. Related Work

Recent advances in 3D reconstruction technology, particularly with the development of Neural Radiance Fields (NeRF) and Gaussian Splatting, have launched a paradigm shift within the subfield of computer vision dedicated to reconstruction. Introduced by Mildenhall et al. [6] in 2020, NeRF presents a novel way to synthesize new views of complex scenes from a sparse set of input images by modeling the volumetric scene function with a fully connected deep network. This method has sparked great interests within the field by demonstrating the ability to produce photorealistic results even in the presence of occlusions and view-dependent effects, which have traditionally challenged other reconstruction techniques. However, despite its advancements, NeRF struggles with real-time applications due to the computationally intensive nature of its forward pass, which involves sampling hundreds of points along rays and querying the neural network for each. Moreover, NeRF's performance degrades on reflective surfaces as it typically interprets them incorrectly as part of the geometry rather than light interacting with the surfaces. Therefore, due to these limitations we focus our project on the other more recent pivotal work Gaussian Splatting by Kerbl et al. [4] These methods handle specular reflections better by combining volumetric Gaussian ellipsoids with spherical harmonics functions to model the radiance distribution over all viewing angles, allowing for more effective rendering of surfaces like those found on cars. Recent works by Jiang et al. [3] and Ye et al. [8] exemplify this approach by enhancing the rendering pipeline with a neural reflection model to better categorize and capture distinct non-Lambertian material properties without sacrificing the training and rendering efficiency. Meng et al. [5] directly tackle the accurate representation of mirror reflections by incorporating additional mirrored viewpoints into the training framework, and Feng et al. [2] integrate physically based rendering to augment the dynamic surface reflections on fluids in the scene.

The inherent ambiguity when projecting 3D information onto 2D planes presents a nontrivial challenge that these new methodologies are beginning to overcome, thereby expanding the boundaries of what can be visualized and interacted with in virtual and augmented reality settings. This is why this project specifically aims to explore how well Gaussian Splatting can address these challenges, particularly in handling complex and reflective surfaces in the automotive context.

3. Background

The core objective that this project tackles is to evaluate and extend effective techniques for reconstructing 3D models of automobiles from a set of photographic views. A key motivation for this work is to accommodate the typical user experience involving modern smartphones to capture "in the wild" vehicles. Considering that most users will not spend more than 3-5 minutes on the capturing process, and often do not possess the steady hands required for professional-quality data collection, our approach must be robust against noise and motion blur. Moreover, the reconstruction workflow should be accessible and feasible on computational resources that are commonly available to consumers, or at least commercially available without being prohibitively expensive.

The project's main technical difficulty is to efficiently find correct correspondences of 3D points across various RGB images that capture a given subject and correctly place them in 3D space, as well as efficiently representing them so that viewing from both original and novel angles is enabled. We primarily investigate two methodologies: COLMAP and Gaussian Splatting. COLMAP is a well-established photogrammetry software that utilizes robust algorithms for Structure from Motion (SfM) and Multi-View Stereo (MVS), providing a foundational approach for generating 3D models from photographic images. On the other hand, Gaussian Splatting represents a more recent advancement in the field, building on top of COLMAP and offering an innovative approach. Next we briefly discuss the relevant mathematical formulations of the two technique.

3.1. Mathematical Background

Fundamentally, COLMAP uses a triangulation model based on the principles of projective geometry to estimate the positions. The process involves identifying key points across multiple images and using these points to reconstruct the positions in a three-dimensional space.

Before triangulating points in 3D, COLMAP first estimates both intrinsic and extrinsic camera parameters to accurately reconstruct scenes. The intrinsic parameters, which include the focal length, skew and distortion coefficients, are crucial for correcting camera and lens distortions and for converting pixel coordinates into camera coordinates. Extrinsic parameters, describing the camera's position and orientation in space, are determined through the process of image registration and alignment. Next, COLMAP performs the geometric verification step, which involves checking the geometric consistency of matched features across multiple images. COLMAP uses the Fundamental or Essential matrix to validate matches based on the principles of epipolar geometry. Note that this step is crucial to filter out erroneous matches that could degrade the quality of the 3D reconstruction, and the reflective automotive surface can be especially susceptible to inaccurate pixel-level matching.

Once initial matches and rough camera poses are established, COLMAP employs bundle adjustment to refine these estimates by minimizing the reprojection error. The reprojection error is the distance between observed feature points in the image and the projection of the corresponding 3D point estimated by the current camera parameters. Mathematically, this is an optimization problem expressed as:

$$\min_{\mathbf{C},\mathbf{X}} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \omega_{ij} \|\mathbf{x}_{ij} - \pi(\mathbf{X}_i, \mathbf{C}_j)\|^2 \tag{1}$$

where $\mathbf{C}_j$ are the camera parameters, $\mathbf{X}_i$ is the i-th 3D point from the j's camera perspective, $\mathbf{x}_{ij}$ are the observed 2D points, π represents the projection function, and ω_{ij} are the weights indicating the reliability of observations. To further improve the robustness of the reconstruction, COLMAP incorporates outlier filtering mechanisms such as RANSAC (Random Sample Consensus) during the initial feature matching phase, as well as during bundle adjustment.

The other subject of our investigation is Gaussian Splatting, and it represents an advanced technique in the field of point-based rendering, that addresses several limitations in traditional photogrammetric and volumetric methods. This approach focuses on rendering point cloud data—an unstructured and discrete geometric representation—by enhancing their representation through ”splatting.” Splatting involves rendering point primitives with a volumetric object larger than a pixel, typically using shapes like circular or elliptical discs, to cover gaps and reduce aliasing issues inherent in simpler point sample rendering methods.

The mathematical model behind Gaussian Splatting aligns with that used in volumetric rendering techniques such as NeRF but simplifies the computation and improves flexibility. The color C along a ray cast during rendering is calculated as:

$$C = \sum_{i=1}^{N} T_i \alpha_i c_i, \tag{2}$$

where: N is the number of points, σ_i is the density at each point, δ_i is the interval between samples along the ray, c_i is the color at each point, $\alpha_i = (1 - \exp(-\sigma_i \delta_i))$ is the opacity of each point, and $\prod_{j=1}^{i-1}(1-\alpha_j)$ is the accumulated transmittance to point i. This model provides a continuous representation by blending ordered points along the ray, where each point's influence is modeled with a 2D Gaussian function related to its opacity and position adjustments. This approach mitigates some of the computational intensity and noise issues associated with NeRF's random sampling by utilizing a discrete, unstructured point setup that can dynamically adjust geometry through optimized opacity and positions. This flexibility is crucial for achieving high-quality reconstructions, particularly when dealing with dynamic scenes or objects with complex reflective properties.

Finally, it should be noted that for the CarWhale360 dataset [7] previously mentioned in the Project Proposal, while it provides a controlled environment for testing, does not align perfectly with the project assumptions of a typical capture scenario. This dataset features cars on a rotating platform captured from a single fixed viewpoint, which contrasts with our need for data captured from multiple angles around a stationary vehicle. After some testing, we determined that the vanilla implementation of Gaussian Splatting is incompatible with the dataset, which highlights Gaussian Splatting's limitation in capturing dynamic scenes.

4. Approach

In this section we discuss the hardware resources and software implementations that underlie the technical exploration of this project.

4.1. Capture

The raw input for the reconstruction algorithms is an HD (1080px by 1920px) video captured on an iPhone 15 Pro using the built-in ultrawide angle camera, which, according to the manufacturer, has a 13 mm focal length, $f/2.2$ aperture and 120° field of view. The videos are captured at 120 seconds long and 60 fps in order to achieve minimal motion blur in individual frames and a comprehensive coverage in a half-dome shape surrounding the subject. From the 7,200 total frames available, we preprocess to select the best 200 frames based on quality metrics.

Specifically, to filter out the blurry frames, we employ a Laplacian variance metric, which quantifies image sharpness. The Laplacian operator measures the second spatial derivative of the image intensity, effectively highlighting regions of rapid intensity change, such as edges and boundaries. The variance of the output is then calculated, providing a scalar value that reflects the degree of spread of the Laplacian in the image. Higher variance values indicate sharper images because they correspond to greater local intensity changes across edges and boundaries, which are characteristic of well-focused images. We incorporate the metric in a simple Python script so that it runs efficiently and seamlessly with the rest of the reconstruction pipeline.

These frames undergo further processing in COLMAP to establish baseline 3D models, serving as foundations for refinement through Gaussian Splatting. Note that the number 200 is chosen based on the trade-off between quality and processing speed, since scenes from 360-degree view synthesis datasets typically contains between 100-300 frames.

4.2. Reconstruction

The reconstructions are performed on a high-performance PC equipped with an AMD Ryzen 7 5800X3D CPU and an Nvidia RTX 3080 GPU with 10GB of VRAM, running Windows 11. This setup provides a stable platform for running the code-base involved in both the COLMAP and Gaussian Splatting reconstruction processes, allowing for convenient installations of the relevant environment and prerequisite packages required by photogrammetry and 3D modeling software.

4.3. COLMAP Workflow

The project primarily utilizes COLMAP's sparse reconstruction process, which involves feature extraction, feature matching, geometric verification, motion recovery, triangulation and outlier removal. The goal of this phase is to focus on efficiently estimating camera poses and creating a sparse 3D point cloud from the input images. The workflow can be quite demanding on computational power, but utilizing modern CUDA-aware optimizations available in the latest

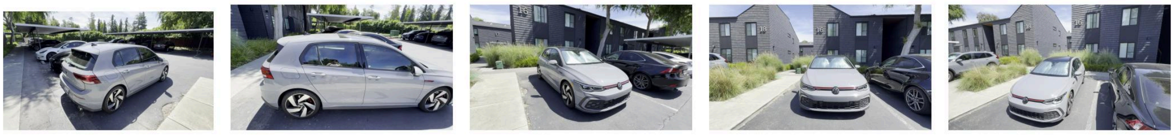


Figure 1. Volkswagen Golf Data From 5 Viewpoints in 45 Degree Increments

versions of COLMAP, we were able to complete an average processing session within around 10 minutes.

Given the success of the sparse reconstruction processing, we also experiment with dense reconstruction as well to generate a more complete picture - a 3D mesh. However, this process is significantly more computationally demanding. Indeed, our tests indicate that dense reconstruction is considerably slower on the existing hardware, which greatly limit its feasibility for this project. Here we only selectively perform dense reconstruction in order to assess its practicality. We visually compare the output at a high level to the Gaussian Splatting result.

4.4. Gaussian Splatting Workflow

After importing the video file from the capturing session, we use a preprocessing script to first filter out low-quality frames, targeting frames that exhibit significant blurring or noise. This ensures that only the best-quality data is forwarded into the reconstruction process, which is critical for achieving high quality in the final 3D models.

For the reconstruction, we leverage the official implementation of Gaussian Splatting available from the GitHub repository by Kerbl et al. [4] The training loop is conducted for 7000 iterations - a checkpoint chosen by the original authors to balance out computational demands and the quality of the output. The output is then both non-interactively rendered from held-out viewpoints for testing and metric evaluation, and interactively rendered using the SIBR viewer software included within the official implementation - enabling real-time, full degree-of-freedom inspection with the reconstructed models. We conclude that this workflow directly taps into a well-maintained and well-documented implementation while also ensuring compatibility to the practical computational limitations faced by this project's consumer-grade hardware.

5. Experiments

5.1. Data Collection

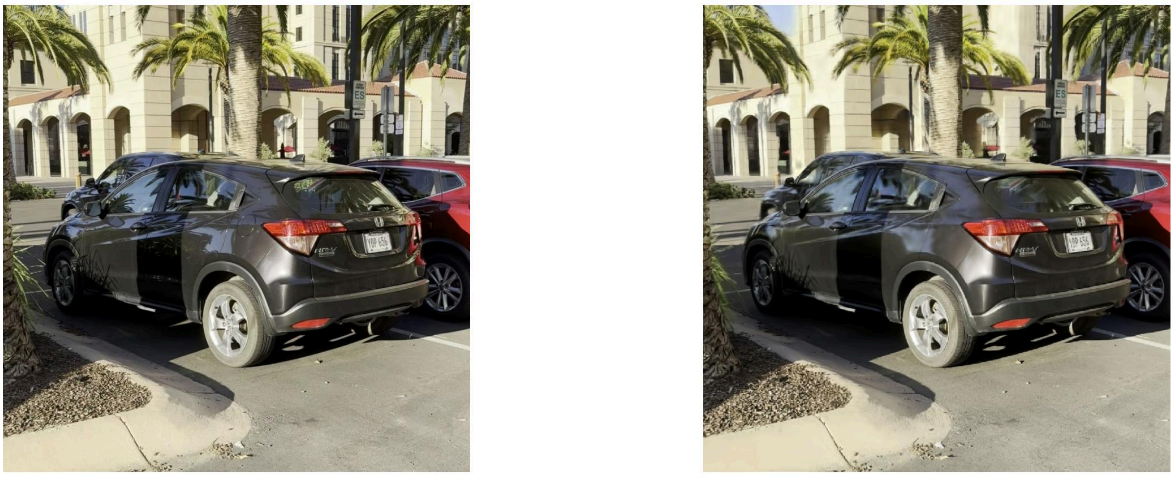
For the first half of the project, we focused on acquiring a custom dataset of various cars around the Stanford campus to evaluate and adjust our methodologies. Specifically, the car models we captured include a diverse range of automobiles in different sizes, design styles, surface features and surface complexities. Our dataset include car models from the following manufacturers: Audi, Honda, Volkswagen, Mazda, Porsche, Tesla, and BMW. We made these choices in order to better represent different color, designs and surface complexities commonly found in urban settings. Figure 1 shows the captured data of a Volkswagen Golf vehicle in a light-gray non-metallic paint from a selection of viewpoints. Additional samples from the collected data are available in the appendix.

5.2. Qualitative Evaluation

To illustrate the effectiveness of the Gaussian Splatting reconstruction techniques, we begin with a detailed side-by-side comparison of the visual outputs for the Honda vehicle, including the sparse and dense reconstruction produced by COLMAP. See Figure 2 and 3.

In the side-by-side visual comparison of the reconstruction results for the Honda, we see several distinct aspects of each methodology's strengths and weaknesses. The sparse reconstruction effectively captures a great number of correspondence key points, particularly around areas of the vehicle where the surface features diverse patterns and distinct structural details, such as the tires, tail lights, and edges. This suggests that sparse reconstruction excels in regions with clear, identifiable features to anchor the reconstruction process. Meanwhile, the dense reconstruction provides an even better representation of the overall shape of the vehicle, filling in many of the gaps that sparse reconstruction might leave. However, it struggles in areas of heavy shadow, as well as smooth, homogeneous surfaces where there are fewer features to anchor the reconstruction process. These areas often appear undefined or inaccurately modeled, indicating the limitation of dense reconstruction in handling scenes with low texture or complex lighting conditions.

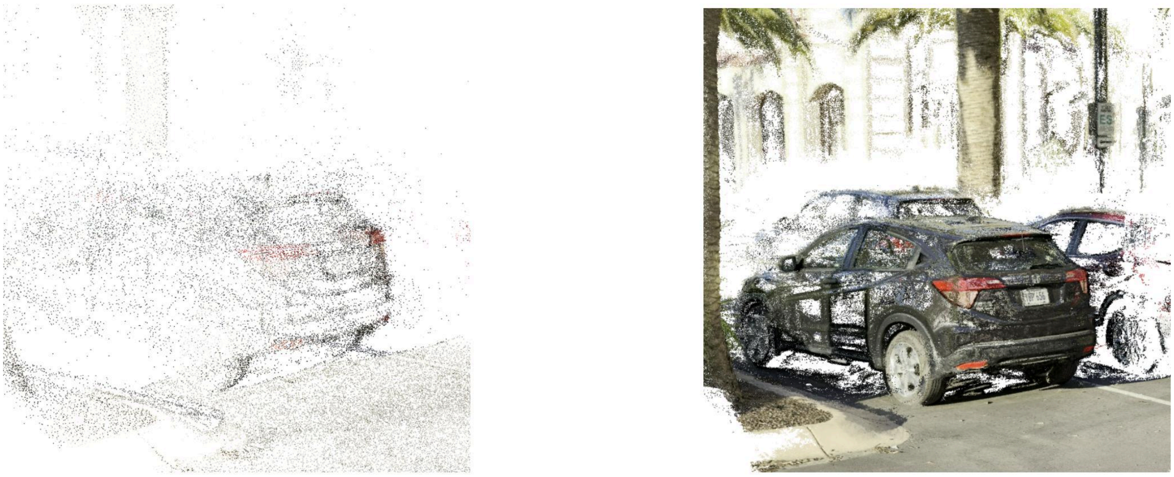
Gaussian Splatting, on the other hand, offers a more comprehensive and cohesive model of the vehicle. This technique manages to paint a complete picture, effectively filling in gaps and holes and robustly handling both specular highlights and deeply shadowed areas. The continuity and smoothness in the Gaussian Splatting output underscore its capability to integrate the visual data more smoothly and thoroughly than the previous two methods. Despite these strengths, it is notable that certain fine details are still missing from the reconstruction, such as the circular gap of the fuel cap to the left of the left tail light. This shows Gaussian Splatting's inability to capture certain extremely fine



(a) Ground Truth

(b) Gaussian Splatting

Figure 2. Comparison of ground truth and Gaussian Splatting results for a Honda HR-V.



(a) Sparse Reconstruction

(b) Dense Reconstruction

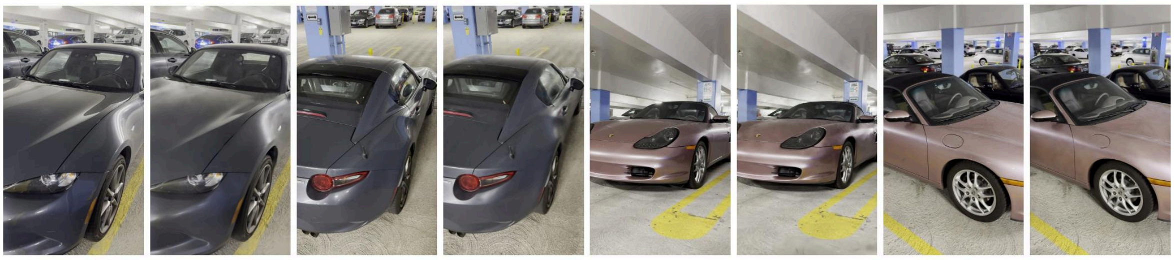
Figure 3. Comparison of sparse reconstruction and dense reconstruction results.

details, which may be smoothed over or lost amidst broader surface modeling.

In Figure 4 and 5 we include some additional side-by-side comparison results of the Gaussian Splatting reconstruction on other vehicles we captured: two in a dark indoor environment, and two in a bright outdoor environment. The figures display ground truth and rendered images from held-out novel viewpoints distinct from the training viewpoints. Notably, here we see that in dark, indoor environments such as a garage the reconstruction quality suffer greatly, due to the lack of exposure of surfaces indirect to the light source and the addition of sensor noise inherent in captures in low-light environment, resulting in the loss of the higher frequency specular details.

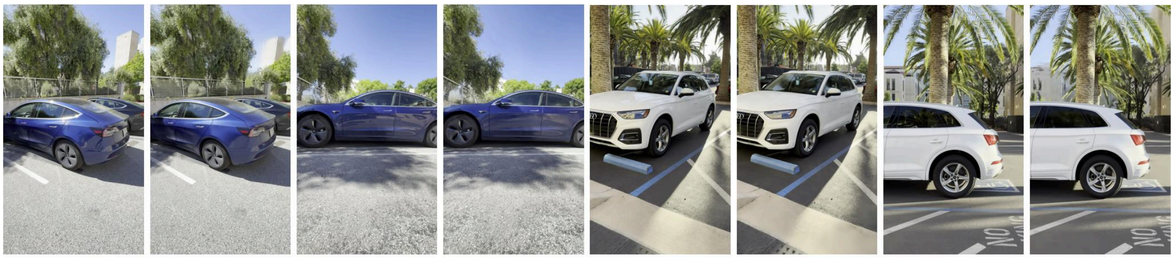
5.3. Quantitative Evaluation

In this section, we present and discuss the numerical results obtained from our 3D reconstruction experiments using three common metrics: SSIM, PSNR, and LPIPS. To give a brief recap: SSIM (Structural Similarity Index) measures the similarity between two images. It considers changes in structural information, luminance, and contrast; PSNR (Peak Signal-to-Noise Ratio) evaluates the ratio between the highest power of a signal and the power of the



(a) GT (b) Render (c) GT (d) Render (e) GT (f) Render (g) GT (h) Render

Figure 4. Mazda Miata and Porsche Boxter Comparisons in a Dark Indoor Environment



(a) GT (b) Render (c) GT (d) Render (e) GT (f) Render (g) GT (h) Render

Figure 5. Tesla Model 3 and Audi Q7 Comparisons in a Bright Outdoor Environment

Car Model	SSIM↑	PSNR↑	LPIPS↓	Training Time
Tesla	0.836	25.94	0.205	540m21s
Audi	0.838	25.02	0.225	40m35s
Honda	0.812	24.20	0.197	245m20s
Mazda	0.891	27.86	0.220	7m20s
Porsche	0.913	29.04	0.169	7m41s
Volkswagen	0.916	30.06	0.160	5m34s
Multi-cars	0.950	29.83	0.161	5m04s

Table 1. Quantitative results of 3D reconstruction for different car models using SSIM, PSNR, and LPIPS metrics, and the corresponding training times.

Output	SSIM↑	PSNR↑	LPIPS↓	Training Time
Average-7K	0.879	27.42	0.191	5m–9h
Author-7K	0.770	25.60	0.279	6m25s
Author-30K	0.815	27.21	0.214	41m33s

Table 2. Quantitative results comparing the average results on the custom car dataset versus 3DGS's original author's results on Mip-NeRF360 for 7K and 30K iterations.

corrupting noise that affects the quality of its representation; LPIPS (Learned Perceptual Image Patch Similarity) assesses the perceptual similarity between images using a trained deep neural networks.

The results from Table 1 show that the reconstruction scores were highest for the Multi-car indoor scene, which contains three models of cars positioned immediately next to each other, and the Volkswagen Golf outdoor scene, achieving the highest SSIM and PSNR values and the lowest LPIPS scores. This suggests that the two crucial factors in attaining superior reconstruction quality are capturing well-exposed, in-focus images and having an abundance of feature point correspondences. Additionally, the training times were generally predictable, except for certain models that took orders of magnitude longer. This unreasonably extended training time is likely due to the 10GB VRAM limitation of the RTX 3080 GPU in conjunction with a lack of optimization for low VRAM support in the training code, since the original paper recommends a minimum of 24GB of VRAM, available in the RTX 4080 series.

Furthermore, Table 2 demonstrates that at 7K training iterations our results perform comparably with the Mip-NeRF360 [1] results from [4] at 30K iterations. We believe one reason for this is likely due to the relatively small amount of 175 training frames (25 held-out for testing) per vehicle compared to the 330 frames per scene in Mip-NeRF360. Another potential reason might be because Mip-NeRF360 contains large unbounded scenes with a central subject that can contain lots of thin structures and intricate geometry more complex than the average vehicle.

6. Conclusion

In this paper, we explored the application of Gaussian Splatting for the 3D reconstruction of vehicles from limited photographic inputs. Our results demonstrate the viability of this method in producing high-quality 3D models, particularly when dealing with the complex and reflective surfaces typical of automobiles. The overall specularity and distribution of highlights and shadows were well-captured by the vanilla 3DGS algorithm, showcasing its effectiveness in rendering smooth and continuous surfaces.

One of the significant achievements of this project was demonstrating that Gaussian Splatting can achieve high fidelity reconstruction results at home using consumer-grade hardware. This approach offers a promising alternative to traditional photogrammetry techniques, especially in handling specular reflections and ensuring accurate geometric representations. The reconstruction quality observed in our custom dataset confirms the potential of Gaussian Splatting in real-world scenarios.

Despite these successes, the project faced several challenges that highlight areas for future improvement. The primary challenge was the demand for powerful computing resources. The training times and memory requirements often stretched the limits of our hardware capabilities even though it was feasible. Future work could explore optimizing the algorithm for lower-resource environments or leveraging more advanced computing resources to reduce training times and enhance performance. Additionally, we explored the GaussianShader project's codebase [3] using WSL2 on our Windows machine. However, this approach further degraded performance and also revealed a lack of support for custom dataset creation in said project.

Additionally, the extreme level of lighting details often missed in the reconstructions was a notable limitation. This could be attributed to the resolution of the input data or the number of Gaussian nodes used during training. Future research could focus on improving the lighting function representation model and further optimizing the number of Gaussian nodes used to capture finer details and more intricate lighting effects.

In conclusion, this project successfully applied Gaussian Splatting to the 3D reconstruction of vehicles, demonstrating both the potential and limitations of the approach. By addressing the challenges of computational demands and improving the data capture setup, future research can build on these findings to develop more robust and efficient methods for 3D reconstruction in various practical applications.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields, 2022. 6

[2] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, et al. Gaussian splashing: Dynamic fluid synthesis with gaussian splatting. *arXiv preprint arXiv:2401.15318*, 2024. 2

[3] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces, 2023. 2, 7

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 2, 4, 6

[5] Jiarui Meng, Haijie Li, Yanmin Wu, Qiankun Gao, Shuzhou Yang, Jian Zhang, and Siwei Ma. Mirror-3dgs: Incorporating mirror reflections into 3d gaussian splatting. *arXiv preprint arXiv:2404.01168*, 2024. 2

[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, 2020. 1

[7] Rszengin. Carwale-360 dataset. https://github.com/rszengin/carwale-360-dataset, 2018. GitHub repository. 3

[8] Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection, 2024. 2

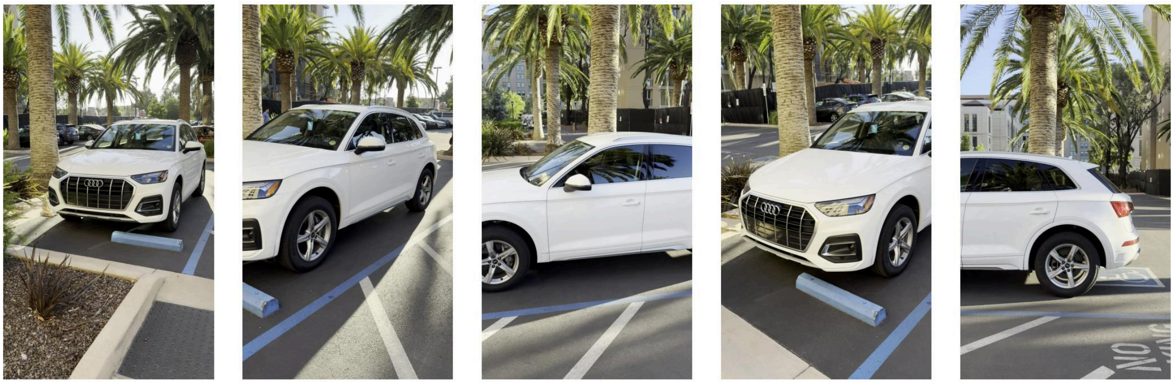


Figure 6. Audi Dataset

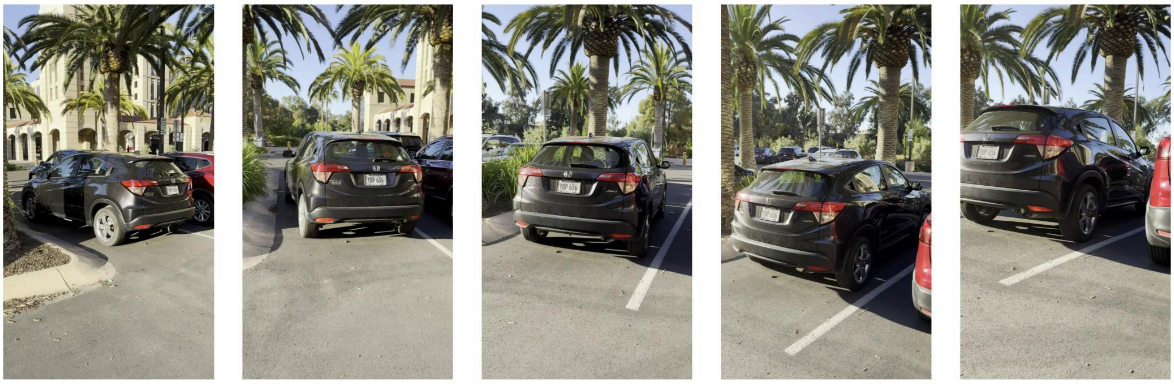


Figure 7. Honda Dataset

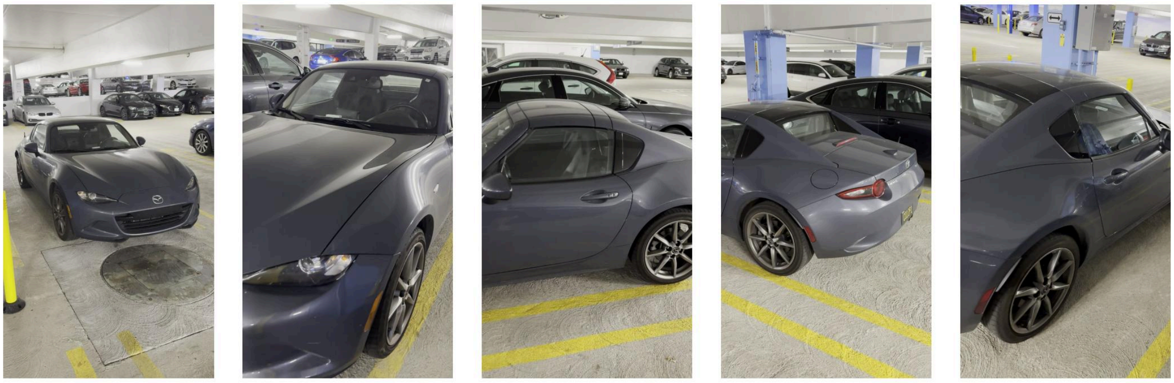


Figure 8. Mazda Dataset

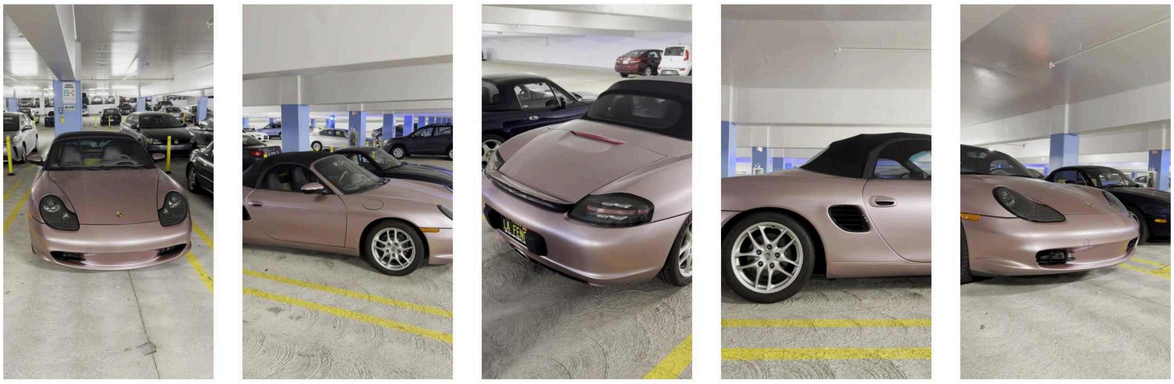


Figure 9. Porsche Dataset

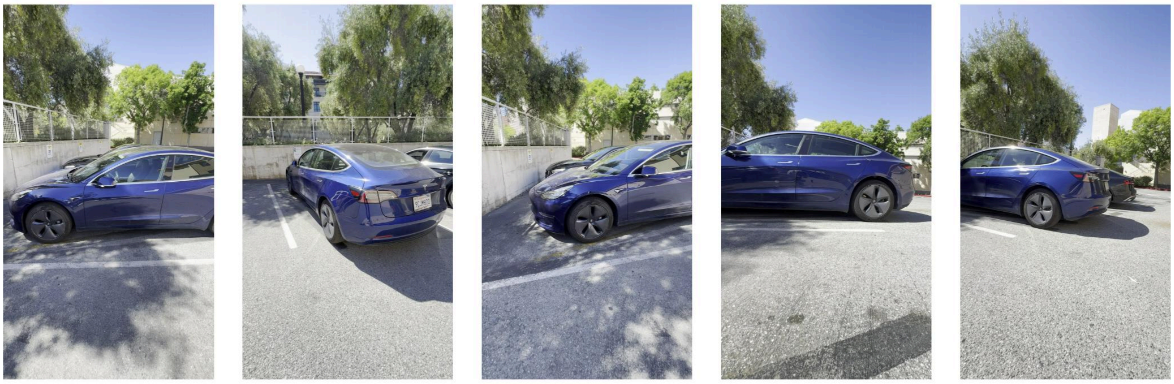
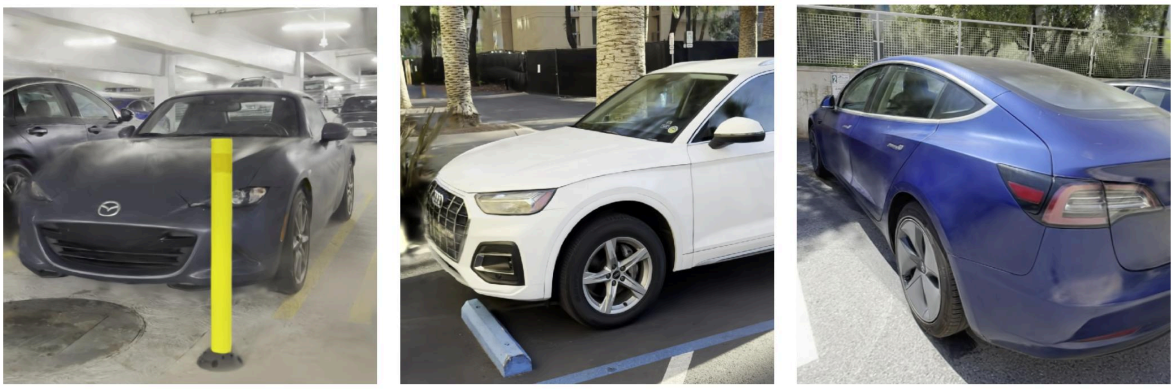


Figure 10. Tesla Dataset



(a) Mazda Miata (b) Audi Q5 (c) Tesla Model 3

Figure 11. Gaussian Splatting reconstruction results from novel viewpoints for each model: Mazda Miata, Audi, and Tesla.