Exploring the Integration of Gaussian Splatting in Formula 1 Racing

Harviel Arcilla
Stanford University
karcilla@stanford.edu

Colette Do
Stanford University
cbdo@stanford.edu

Abstract

This paper investigates the intersection of Formula 1 data and 3D Gaussian Splatting (3DGS), examining the opportunities and limitations of applying 3DGS techniques to the fast-paced, visually complex setting of Formula 1 racing. While prior work has applied 3DGS to general automotive environments, its effectiveness in high-speed motorsport contexts remains unexplored. We compare the standard 3DGS pipeline built on COLMAP with a globally optimized GLOMAP-based alternative to evaluate differences in robustness and efficiency. Leveraging onboard footage, telemetry, and multi-angle perspectives, we analyze reconstruction quality, pose accuracy, and the impact of extreme motion on performance. Our experiments include telemetry-based pose initialization, camera parameter estimation under varying visual conditions, and multi-view training using virtual 360 data. Ultimately, we assess the feasibility of using 3DGS for practical applications in Formula 1, including crash scene reconstruction and analysis of off-track excursions, using only readily available telemetry and onboard video.

1. Introduction

Recent advancements in neural rendering techniques, particularly 3D Gaussian Splatting, have opened new frontiers for real-time and photorealistic scene reconstruction. In this work, we explore the potential of applying Gaussian Splatting to Formula 1 racing footage—a domain characterized by extreme speeds, rapid motion, high-frequency vibrations, and complex visual environments. Formula 1 presents a uniquely demanding testbed for evaluating the capabilities and limitations of current scene reconstruction methods. By leveraging onboard footage, synchronized multi-angle views, and real-world telemetry data, we assess how Gaussian Splatting performs under high-speed motion, its accuracy in representing track geometry, and its potential utility in complex visual contexts such as during crash events. In summary,

- We analyze two methods of producing Structure-from-Motion objects. Our pipeline takes in sequential frames of racing footage as input, processes the footage into camera poses using an Structure-from-Motion method, and then reconstructs the scene using Gaussian splatting.
- We explore using telemetry data as input and output an interpolated list of camera poses that can be used as input or evaluation.
- We investigate Gaussian splatting's ability to estimate and analyze onboard camera parameters, even with distortion due to heat and vibration, through experiments. For this experiment, cropped videos are used as input to the 3D Gaussian splatting procedure, and a reconstructed scene is the output.
- We experiment with multi-view cameras for improving reconstructions. We use a 360deg camera as input to the 3D Gaussian splatting procedure and output a reconstructed scene.

Our aim is not only to evaluate this technology's technical performance in extreme conditions but also to highlight its practical applications in motorsport analysis, simulation, and safety research.

2. Related Works

2.1. Structure-from-Motion

Early research on matching images for geometric reconstruction has explored different techniques. One proposed method was unguided matching, a procedure that tries to search for image matches within a corresponding region in other images for a particular point. Other methods optimize for some geometric constraint where image points are considered matches or inliers for a particular feature if they satisfy the specified geometric constraint [1]. For these types of optimizations, random sample consensus (RANSAC)–an algorithm that involves sampling sets of points and using the rest of the points to test for inliers for a set number of

iterations–is widely used to estimate camera parameters or image matches [3].

Structure-from-Motion (SfM) is a method that infers scene geometry by matching points across images of differing camera positions. While this method can produce sparse point clouds, SfM does struggle to produce complete and robust results. COLMAP improves on the naive SfM method by implementing a geometric verification strategy, "next best view" selection, triangulation method, bundle adjustment, and an outlier filtering strategy that utilizes RANSAC [10]. The improvements introduced by COLMAP allow for more stable reconstruction due to its iterative reconstruction process. More recently, GLOMAP, Global Structure-from-Motion, has been introduced using descriptor-based methods and global estimations [7]. While COLMAP iteratively estimates camera positions, GLOMAP estimates the position of images all at once by using rotation and translation averages. GLOMAP performs on par with or exceeds COLMAP's performance and requires less time. However, GLOMAP does tend to have limitations with symmetric objects during its rotation averaging step [7].

Both COLMAP and GLOMAP, state-of-art models or foundations for SfM, are integral to our pipeline, serving as preprocessing tools for camera pose estimation and sparse geometry extraction. The efficiency and scalability of GLOMAP, in particular, are crucial for enabling real-time scene reconstruction in fast-paced sports environments.

2.2. 3D Gaussian Splatting

SfM objects often act as inputs to algorithms that can utilize the feature mappings to generate 3D reconstructions. One method is 3D Gaussian Splatting, which offers a fast and visually compelling method for scene rendering, leveraging a point-based representation augmented with anisotropic Gaussian kernels [6]. Each point in the reconstructed scene is modeled as a 3D Gaussian with defined mean and covariance, projected onto a 2D screen as a "splat." These splats are then rendered using α-blending in back-to-front order. This method is much faster than previous Neural Radiance Field techniques and is competitive in quality with other rendering methods, but it does struggle with scenes that have not been properly observed, leading to elongated artifacts [6].

Since its release, other research has attempted to augment 3DGS with other capabilities and tasks, specifically automotive tasks. SplatAD was introduced to allow for camera and LiDAR pose estimation for autonomous driving tasks [5]. LiDAR positions were rendered similarly to normal image rendering, except the 3D gaussians generated were converted into spherical coordinates; optimizations used the same loss function as normal 3DGS with the addition of LiDAR intensity and depth [5]. SplatAD improved runtime, but the method is limited to rendering only rigid objects, approximating dynamic actors as static. Another work uses composite gaussian splatting to allow the modeling of dynamic actors. DrivingGaussian uses incremental 3DGS to reconstruct the background, then overlays a composite dynamic Gaussian splatting graph that models dynamic objects [14]. LiDAR data was used as a validation metric to ensure geometric consistency. Within the automotive field, prior work has also explored how 3DGS can be improved by removing the SfM generation process. One project explored using global track constraints instead of COLMAP to improve the geometric consistency among views [11]. Instead of trying to estimate camera poses, the method uses the tracks from matching features in all of the images to optimize camera parameters. This method improves on the original 3DGS pipeline by handling complex camera views more accurately in reconstruction.

Our project seeks contribute to the applications of 3DGS in the automotive and motorsport field by reconstructing tracks with fast-moving vehicles. Building on ideas of augmenting 3DGS with LiDAR, we aim to improve the model's robustness using telemetry data that are already collected by racing vehicles.

2.3. Motorsport Visualization Tools

Data collection systems in motorsports have long recorded speed, performance, and driver inputs for further analysis by engineers. Tracks also usually have various cameras at different viewpoints and parts of the track for teams and regulators to review. Traditional visualization tools combine telemetry with 3D modeling software–Visual C++, Open Graphics Library, and CAD tools–to support race strategy and engineering decisions [8]. These visualizations enabled real-time modeling of vehicle trajectory and racing lines, improving safety assessments and driver performance. In addition to providing more visual information, this reconstruction was able to improve trajectory calculations due to more precise racing lines [8]. This data is prevalent within popular motorsports like Formula 1, but smaller, amateur motorsports do not always have access to dense sensor data since they are only collected at specific locations on the track. Work has been done on how interpolation can be used to estimate racing lines between sparse points [12]. Using the predetermined track positions and reference times at specific sectors on the track, the authors are able to produce visualizations that closely model the true vehicle position at smaller scale racing events.

Our project hopes to build on the current 3D modeling techniques used in motorsports to improve driver analysis and even help with crash reconstruction–which has an added element of model/vehicle deformation that is not possible using current methods. Similar to how these techniques used existing forms of data collection, our project at-

tempts to utilize already existing onboard cameras and track camera views to achieve better than baseline results.

3. Dataset and Features

Our dataset comprises of both 22 simulated and real-world segments of racetracks with multi-view racing footage. Real-world clips collected from public footage of professional races [4] and simulated clips were recorded on driving simulators. We focus on well-documented tracks such as Circuit de Monaco and Yas Marina Circuit and consider only a few complex corners of each track. We also collect the telemetry attributes of the laps using FastF1 API [9]. Each clip is approximately 30 seconds long. We preprocess the data by extracting 400-500 frames from each clip–roughly enough to capture a major corner or portion of track. This combination of synthetic and real-world footage allows us to evaluate our model's generalization ability across different lighting conditions, motion patterns, camera setups, and ideal scenarios.

4. Methods

4.1. SfM: COLMAP

COLMAP follows a three-step procedure: feature detection and extraction, feature matching and verification, and incremental reconstruction. During feature extraction, local features that are both radiometrically and geometrically invariant are found [10]. During feature matching and verification, the method tests every pair of images for feature overlap and verifies the mapping. During incremental reconstruction, pose estimates are outputted following image registration, multi-view triangulation, and bundle adjustment. Images are registered once its pose cameras and intrinsic paramters are found, and the image can be triangulated to at least one other image in the model. Bundle adjustment refines the camera $\mathbf{P}_c$ and point $\mathbf{X}_k$ parameters found by grouping together redundant images $\mathbf{G}_r$ and minimizing reprojection error (Eq. 1) where ρ is a loss function and π is a scene to image space projection function. For this method, we used code from the original authors [10].

$$E = \sum_j \rho_j(||\pi(\mathbf{G}_r, \mathbf{P}_c, \mathbf{X}_k) - \mathbf{x}_j||_2^2) \tag{1}$$

4.2. SfM: GLOMAP

GLOMAP is an alternative SfM method that emphasizes efficiency by adopting a global estimation approach rather than COLMAP's incremental method. GLOMAP is built on top of COLMAP's feature extraction and feature mapping procedure. However, instead of incremental reconstruction, GLOMAP employs global estimation and global bundle adjustment, which has the advantage of being initialization free and robust to outliers. Instead of the reprojection error used by COLMAP, the authors developed a different optimization function (Eq. 2).

$$\arg_{\mathbf{X},c,d} \min \sum_{i,k} \rho(||\mathbf{v}_{ik} - d_{ik}(\mathbf{X}_k - \mathbf{c}_i)||_2), \quad \text{subject to } d_{ik} \geq 0. \tag{2}$$

In this equation, $\mathbf{v}_{ik}$ represents the camera ray that denotes the vector of point $\mathbf{X}_k$ from camera $\mathbf{c}_i$ and d_{ik} is a normalization constant [7]. This optimization improves on COLMAP by bounding the errors in the range $[0, 1]$ and can converge with random initialization. For this method, we used code from the original authors [7].

4.3. Baseline: 3D Gaussian Splatting (3DGS)

The baseline Gaussian Splatting model introduced by Kerbl et al. uses COLMAP to create its SfM point clouds and outputs a 3D reconstructed model. This method builds on point-based rendering by representing each 3D point in the scene as a Gaussian, defined by a mean and covariance matrix (Eq. 3).

$$G(x) = e^{-\frac{1}{2}(x)^\top \Sigma^{-1}(x)} \tag{3}$$

For storage purposes, the Gaussian is stored as a scaling three-dimensional vector and a quaternion for rotation [6]. Optimization of these gaussians includes projections and density control such that geometry can be created, moved, or destroyed. The loss function used combines an $L1$ loss term with a data structural similarity index (D-SSIM) term where λ is a constant from 0 to 1 (Eq. 4) [6].

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{4}$$

Rendering is done by a tile rasterizer that uses alpha, or transparency, compositing in back-to-front order, which is fast and efficient for real-time applications [6]. We used the original package from the authors [6].

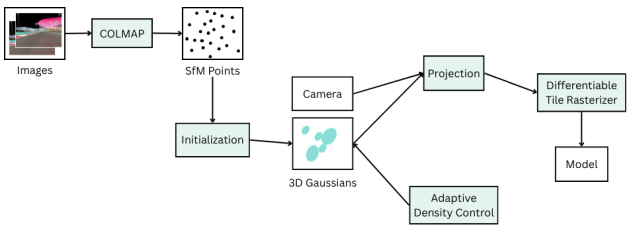

Figure 1. Baseline: Gaussian Splatting

4.4. Global Gaussian Splatting

Our global gaussian splatting (GLGS) method uses GLOMAP to create SfM objects. GLOMAP calls on COLMAP's feature extraction and feature mapping methods, then uses a global reconstruction method and bundle adjustment. Finally, the resulting SfM object is passed to the Gaussian splatting method.

4.5. Telemetry and Pose Calculations

Camera poses for onboard views can be estimated using telemetry data from each Formula 1 car's onboard GPS and vehicle systems. A subset of this data is made publicly available for broadcasts and is accessible through the FastF1 Python library [9], including timestamped global positions. With GPS accuracy within approximately 15 centimeters, the data provides a spatial reference for pose estimation. Orientation is inferred by assuming the car faces its next telemetry point, producing forward-facing vectors representable as quaternions. While useful for initialization, this sequence alone is insufficient for structure-from-motion. Temporal interpolation is needed to align with video frames, and further refinement. For this method, we developed code using the FastF1 library.

4.5.1 Quaternion Pose Interpolation

COLMAP uses unit quaternions $\mathbf{q}$ for representing orientation, allowing smooth rotation interpolation. Given two keyframes $(\mathbf{p}_0, \mathbf{q}_0)$ and $(\mathbf{p}_1, \mathbf{q}_1)$, interpolated pose at $t \in [0, 1]$ is computed as:

$$\begin{aligned} p(t) &= (1-t)p_0 + tp_1 \\ q(t) &= \frac{\sin((1-t)\theta)}{\sin\theta} q_0 + \frac{\sin(t\theta)}{\sin\theta} q_1, \quad \theta = \cos^{-1}(q_0 \cdot q_1) \end{aligned} \tag{5}$$

where $q(t)$ is the spherical linear interpolation (SLERP) ensuring smooth rotational transitions. This combined interpolation results in continuous, smooth trajectories of both position and orientation in real space.

4.6. FOV Estimation from Camera Intrinsics

Camera field of view (FOV) can be estimated from COLMAP's learned intrinsics when using the pinhole camera model. COLMAP optimizes focal length parameters (f_x, f_y), from which the horizontal or vertical FOV can be computed using:

$$FOV = 2 \cdot \arctan\left(\frac{\texttt{sensor\_size}}{2 \cdot f}\right)$$

where f is the focal length in pixels and `sensor_size` is the physical sensor dimension. Despite real-world deviations from the pinhole model, this method yields reasonable FOV estimates, especially for small-sensor cameras common in onboard applications.

5. Experiments, Results, and Discussion

5.1. Evaluation Metrics

We use the following metrics for evaluating our methods and experiments [13].

5.1.1 Peak Signal to Noise Ratio (PSNR)

PSNR measures the quality of the reconstruction by showing how accurately the signal is preserved during reconstruction.

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

5.1.2 Structural Similarity Index Measure (SSIM)

SSIM is a method that can measure the similarity between images using luminance l, contrast c, and structure s when considering pixel sample mean μ, sample variance σ^2, sampling covariance σ_{xy}, dynamic range L, and stabilization values c_1, c_2.

$$\begin{aligned} SSIM(x,y) &= l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma \\ l(x,y) &= \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1};\ c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}; \\ s(x,y) &= \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \end{aligned} \tag{7}$$

5.1.3 $\mathcal{L}_1$ Loss

$\mathcal{L}_1$ Loss, also known as mean absolute error (MAE), calculates the absolute differences between a predicted element $\hat{y}_i$ and the target element y_i.

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{8}$$

5.2. Experiment 1: 3D Reconstruction

5.2.1 Experiment

We used a frame rate of 25 frames per second (FPS) for each racing clip. In preliminary tests, we experimented with using 2, 5, 10, 25, and 50 frames in COLMAP. However, we noticed that with 2, 5, and 10 FPS, COLMAP would struggle to estimate poses for many of the images, while with 50 FPS it would often create disjoined tracks and take too long with compute. While 25 FPS tends to be high as an input for COLMAP, drivers can achieve upwards of 370 km/h, which is over 100 meters traveled in one second. Therefore, each frame shifts two to four meter which we found to be optimal for balancing compute and accuracy.

We begin by detecting and extracting image features using COLMAP's pinhole camera model and default parameter settings. Because the input video frames are sequential in nature, we use a sequential matching strategy to increase correspondence accuracy between temporally adjacent images. For the baseline model, COLMAP was used for incremental reconstruction. For the Global Gaussian Splatting model, GLOMAP was used for global reconstruction,

building off of COLMAP's feature extraction and matching. Gaussian splatting was set for 30,000 steps, as longer training did not significantly improve the reconstruction. For this experiment, we used SSIM, PSNR, and $\mathcal{L}_1$ loss.

5.2.2 Results and Discussion

3DGS-GP	L1 7K	PSNR 7K	L1 30K	PSNR 30K
SGP	0.0345	23.2804	0.0162	29.8112
MCO	0.0482	22.1556	0.0179	29.6134
UAE	0.0233	25.7246	0.0131	31.3956
MIA	0.0910	17.3240	0.0539	21.1638
AUS	0.0279	24.6804	0.0179	28.3987
GLGS-GP	L1 7K	PSNR 7K	L1 30K	PSNR 30K
SGP	0.0491	21.5023	0.0163	30.5745
MCO	0.0441	22.7742	0.0156	30.5889
UAE	0.0216	27.0788	0.0114	33.1589
MIA	0.1381	14.2397	0.0525	20.7815
AUS	0.0488	20.8762	0.0258	25.5903

Table 1. GLGS and 3DGS metrics for Grand Prix (GP) Circuits: L1 Loss and PSNR at 7K and 30K steps

Circuit	3DGS SSIM	GLGS SSIM
SGP	0.5676	0.5811
MCO	0.5923	0.5935
UAE	0.6586	0.6310
MIA	0.4749	0.4588
AUS	0.5798	0.5934

Table 2. GLGS and 3DGS SSIM over 5 frames

The baseline 3DGS model was able to reconstruct track segments with a PSNR around 28 and an SSIM of around 0.57 across five tracks (see Table 1). The GLGS model achieved a PSNR of around 28 and an SSIM of around 0.57 across the five tracks. In other words, the performance of COLMAP with 3DGS and GLOMAP with 3DGS was similar; however, specific tracks show more differences. In Figure 2 Singapore, both methods struggled to reconstruct "Singapore Airlines" on the barriers–likely due to the repetition of the logo throughout various parts of the track– however GLOMAP struggled far more due to its method of global feature matching where the barriers are almost obscured and illegible. For this frame, the 3DGS PSNR was 13.66, and the SSIM was 0.5258. The GLGS PSNR was 11.76, and the SSIM was 0.4786. It is possible that the COLMAP model could not find any suitable location for this specific frame and just chose to rely on repeated track features. On the other hand, GLGS likely could not match this frame globally so very little reconstruction occurred. In Figure 3 Australia (first frame), both models did fairly well visually with GLGS having a slightly higher PSNR and SSIM than 3DGS. While the method did not have as many artifacts in Australia, the resolution of both was worse than the original image. It is also possible that the wider angle allowed both to better reconstruct the beginning of the dataset.

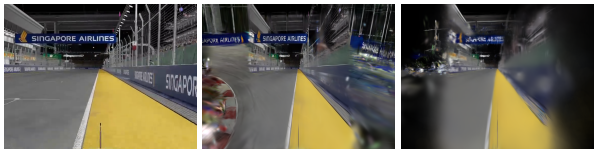

Figure 2. Comparisons between ground truth image (left), 3DGS (middle), and GLGS (right) in Singapore

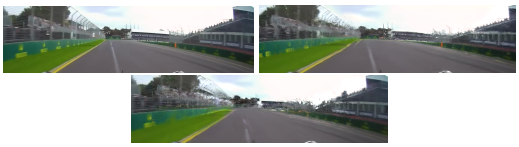

Figure 3. Comparisons between ground truth image (top left), 3DGS (top right), and GLGS (bottom) in Australia

Overall, GLOMAP can provide some additional stability but does not seem to drastically improve the results in this task; in some cases where repetition is prominent, the reconstruction is worse than 3DGS. Notice Figure 4, where COLMAP's reconstruction of Monaco was disjoint compared to GLOMAP, however GLGS lacked detail in complex patterned sections. While this GLGS is more efficient, it is possible that the recurring advertising and equipment in Formula 1 images makes GLOMAP less suitable on some tracks to match features and distinguish between frames (more Reconstruction examples can be found in the appendix).

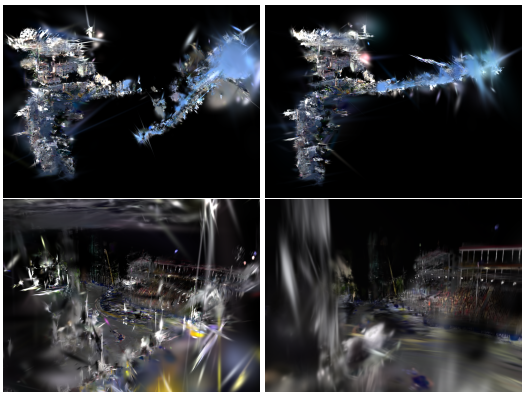

Figure 4. Aerial views of the 3DGS model in Monaco and Singapore (COLMAP Left, GLOMAP Right)

5.2.3 Applied 3DGS Reconstructions

While 3DGS reconstructions are an intriguing technology, their current applications remain somewhat limited; however, they may hold considerable potential within the context of Formula 1 crash analysis. When ample footage is available, 3DGS can be used to generate detailed 3D reconstructions of crash events. These reconstructions may support investigations into the causes and dynamics of incidents, and offer a valuable tool for training marshals, medical teams, and engineers in responding effectively to complex crash scenarios. For this experiment, we examined two complex crash scenarios: George Russell's incident at the 2024 Australian Grand Prix and Sergio Pérez's crash during qualifying at the 2022 Monaco Grand Prix. Both events were classified as red-flag conditions due to the immediate compromise of track safety.

Figure 5. Russell Australia 2024, Pérez Monaco 2022 respectively. (Top) Novel view from 3DGS, (Left) Onboard Training Data, (Right) Reference

Using COLMAP and 3DGS, we reconstructed the scenes based on training data captured from the onboard camera of the car directly behind the crash (Figure 5). From this data, we generated novel views from alternative on-track positions, offering new visual perspectives of the incidents shortly after impact—before track marshals or safety personnel had entered the circuit. The reconstructions captured the vehicles and surrounding environment with discernible detail, though we speculate that improvements in camera resolution and lighting conditions would further enhance the accuracy and realism of such reconstructions in future implementations. Overall, we believe in the method has immense potential in further understanding of incidents like these.

5.3. Experiment 2: Evaluation with Telemetry

We examined two different approaches to employing telemetry in tandem with 3DGS for Formula 1–training and evaluating camera poses.

5.3.1 Initialization Experiment

As an initial attempt to improve initialization, we interpolated the GPS telemetry to a suitable refresh rate similar to the video input. We then performed feature extraction and sequential matching on the video frames, aligning the resulting initial poses with the GPS data prior to running reconstruction. However, this approach was ineffective, as inconsistencies and misalignment between the video input and the interpolated GPS data led to poor pose estimation and unreliable reconstruction results, with typically only a few images being utilized in the COLMAP model. (A telemetry-Generated COLMAP can be found in the appendix).

5.3.2 Evaluation Experiment

We additionally utilized GPS telemetry data to assess the accuracy of the camera trajectory estimated by COLMAP. Trajectories derived from COLMAP poses were visualized and compared against ground truth paths obtained from the telemetry, enabling a qualitative evaluation of reconstruction accuracy. Note: Scale discrepancies were not considered in the evaluation, as no scale parameters were incorporated into the model. We also debated on using quantitative methods to compare the paths but found that the initial camera angle was dominating the results (even after aligning them).

5.3.3 Results and Discussion

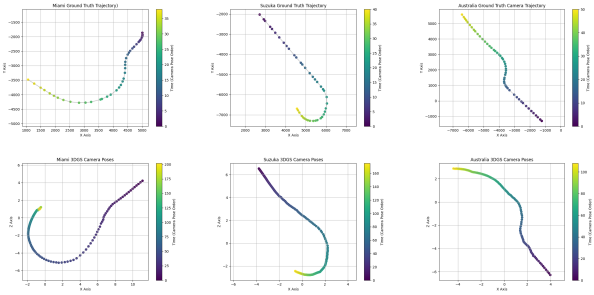

Figure 6. (Top) GPS Ground Truth Route, (Bottom) 3DGS Generated Route

We see that the resulting trajectory and pose graphs closely matched the general shape of the ground truth path, with noticeable errors at the start and end of corners (Figure 6). Unsurprisingly, COLMAP struggled on straights, where cars are traveling the fastest and track-side detail is diminished due to long repeating barriers and scenery. Overall, COLMAPs are not yet effective at modeling track geometry precisely, even if the visual representations may appear accurate from common angles. We noted, however, that 3DGS still performs well on geometry directly on the course.

5.4. Experiment 3: Cropping and Camera Parameters Estimation

The standardized T-camera used across all Formula 1 cars offers a unique opportunity to study the effects of real-world conditions on 3D reconstruction under controlled hardware and placement. Despite identical recording setups, we observed that different sections of identically cropped footage produced significantly varied camera parameters when processed with COLMAP or GLOMAP. This prompted further crop-based experiments to probe how motion and lighting affect reconstruction stability–particularly investigating the sensitivities in current photogrammetry.

For our initial experiments, we evaluated the impact of different training data crops on reconstruction performance. We began with the full-resolution T-camera footage (1920×1080), applying aggressive masking to occlude the car and prevent interference with feature matching. While this approach yielded acceptable results on some circuits, the high memory demands significantly slowed training. Given the high-speed nature of the footage, we also hypothesized that motion blur near the car body would diminish the value of this additional data. We therefore tested a narrower crop (1750×450) focused solely on the forward-facing region, excluding the car entirely (Figure 7). This adjustment substantially reduced training time without any noticeable loss in PSNR or visual quality.

Figure 7. (Yellow) Cropped, (Red) Masked

Although Formula 1 does not publicly disclose the field of view (FOV) specifications of its onboard cameras, 3D model estimate suggest effective FOV close to 90° after cropping, we used this as our comprison to the COLMAP parameters.

5.4.1 Results and Discussion

Using the procedure outlined in the methods section, FOV estimates were derived from COLMAP's learned focal lengths under the pinhole model. The resulting estimates ranged from 30° to 60°, significantly narrower than expected. This discrepancy suggests that COLMAP may compress scene geometry, particularly under challenging lighting or motion conditions. Notably, reconstructions from night-time circuits with strong artificial lighting overhead–such as Singapore and Yas Marina–yielded more consistent and plausible FOV values, likely due to improved feature correspondence in well-illuminated frames. Overall, we found that Formula 1 T-cameras and mountings have significant room for improvement in terms of distortion and noise isolation, but that the high-speed nature of the content may also be to blame in terms of the unreliable camera intrinsics.

5.5. Experiment 4: Multi-view Reconstruction

The use of multiple onboard cameras in Formula 1 cars presents an intriguing opportunity for studying Gaussian Splatting, due to the inherent redundancy and overlap across viewpoints, which can potentially improve 3D reconstruction through better spatial coherence. Our initial exploration focused on leveraging real-world footage, where multiple high-quality onboard perspectives are theoretically ideal for multi-view training. However, in practice, access to synchronized and complete camera views is limited, as not all onboard angles are publicly released, and those that are often suffer from occlusion due to debris, motion blur, and intense vibrations.

5.5.1 Experiment

We initially attempted training on footage from a real 360-degree onboard camera, but encountered significant challenges, particularly with feature matching, which struggled to maintain consistency under such dynamic and visually noisy conditions. This led us to explore an alternative data source: the virtual 360-degree cameras available in the newly release F1 game, EA SPORTS™ F1® 25. We sourced a pre-release virtual 360 video [2], and used it for training (Figure 8). From the 360 video, front, left, rear, and right views were samples (excluding the car from the frame), crucially identical virtual cameras settings were used (Figure 9).

5.5.2 Results and Discussion

With the virtual data, distortion and noise from vibrations and debris were mostly eliminated, providing a good foun-

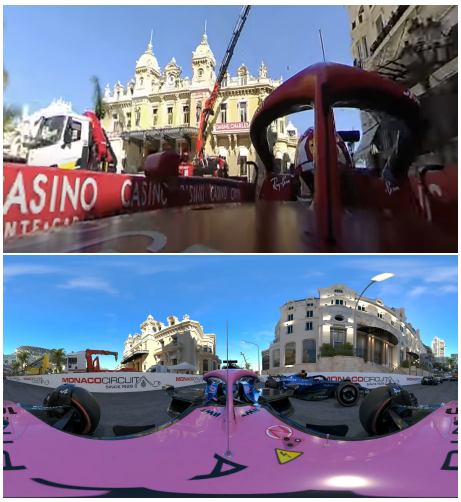

Figure 8. (Top) Real 360 Frame, (Bottom) Virtual 360

Figure 9. Sampled Views from 360

dation for features. Feature extraction was done with shared camera intrinsics (since they shared the virtual specs) and we tried two approaches for feature matching: sequential and exhaustive matching. While sequential converged more quickly, it would rarely utilize all training images due to lack of feature pairs across video feed, we found exhaustive searching with `buffer_size` = `sequence_length` to be optimal for convergence time and training data utilization, since images of the same camera were grouped together. Under these settings, COLMAP returned the densest points-per-image of any of the reconstructions we created (Figure 10).

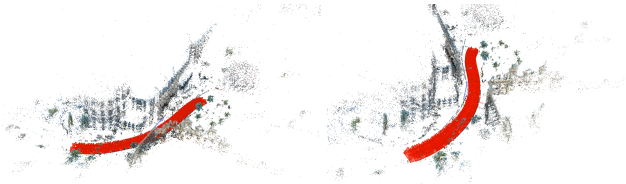

Figure 10. COLMAP Representations

The resulting 3DGS of the multiview training set was a significant qualitative improvement over all other examples we computed (Figure 11). Complex geometry was retained with street lamps and posts being well represented. Colors and distance were also preserved. Overall, we think this is an exciting avenue for Formula 1 to explore, as current reconstructions are extremely coherent with potential for improvement with greater camera resolution and more training data.

Figure 11. (Multi-View) Off-track Novel 3DGS Views

6. Conclusion

In this work, we explored the feasibility and effectiveness of applying 3D Gaussian Splatting (3DGS) to the dynamic and high-speed environment of Formula 1. By comparing the traditional COLMAP-based 3DGS pipeline with a globally optimized GLOMAP approach, we demonstrated potential advantages and drawbacks between using iterative and global feature techniques and their potential in crash analysis. Using real-world onboard footage and GPS data, we explored the effectiveness of of using telemetry for pose estimation and assessment. While exploring various environments, we analyzed camera parameters and behavior under motion. Lastly, with virtual multi-angle footage, we demonstrated the potential for rich, single-pass reconstructions possible through greater camera technology. Our results show that with minor adaptations, 3DGS can be a powerful tool for motorsport applications—particularly in reconstructing crash scenes and analyzing track events using existing video sources.

Due to substantial dependency issues with COLMAP and Gaussian splatting, with more time, we would want to explore how we can augment GLOMAP with the telemetry data and even use it in optimization. Similarly, if we had more resources and access to more granular GPS data, we would also consider using the interpolated positions for Gaussian splatting, removing the need for COLMAP, which could improve efficiency in real-time applications. Since we saw potential in multiview reconstruction, we would also like to explore using footage from multiple cars to improve the reconstruction. Nevertheless, we believe that this work opens the door to further research on integrating learned priors and real-time processing in high-motion environments.

7. Appendices

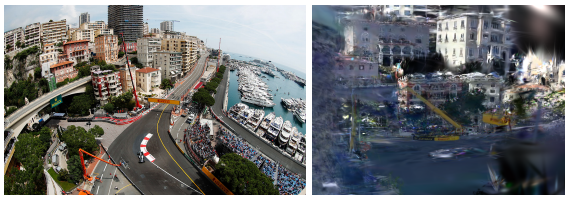

Figure 12. (Left) Sainte Devote Corner, (Right) Sainte Devote GLOMAP

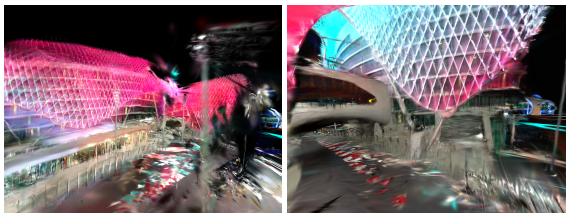

Figure 13. Yas Marina GLOMAP GLGS

Figure 14. Novel COLMAP 3DGS views (Left to Right, Top to Bottom): Miami, Japan, Canada, Australia

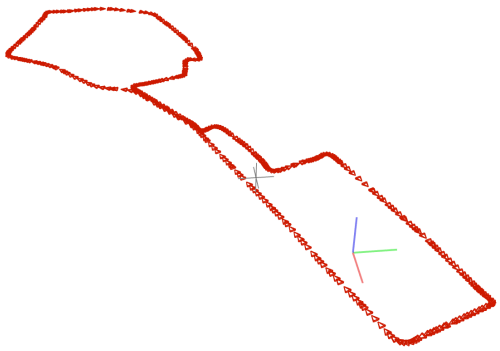

Figure 15. COLMAP generated from GPS telemetry of Azerbaijan Grand Prix Circuit

Figure 16. (Multi-View) More off-track Novel 3DGS Views

8. Contributions and Acknowledgements

Colette contributed to setting up the original implementation of Gaussian splatting, set up GLOMAP, and worked on scoring and evaluation functions. Colette mainly wrote the related works and data sections.

Harviel contributed to setting up the original implementation of Gaussian splatting, set up COLMAP, and worked on the telemetry functions. Harviel wrote the introduction and conclusion sections.

Both contributed to writing methods and results. Both also worked on training the models and spent substantial time troubleshooting dependency issues with installing/running the software.

We would also like to acknowledge the work of the authors of COLMAP(https://github.com/colmap/colmap, GLOMAP (https://github.com/colmap/glomap), Gaussian Splatting (https://github.com/graphdeco-inria/gaussian-splatting), and FastF1 (https://github.com/theOehrly/Fast-F1), as their software was integral to our project. We also want to acknowledge Scikit-image for their utility functions (https://github.com/scikit-image/scikit-image. Lastly, we would like to acknowledge Formula 1, as they are the premise and foundation of the data for the project.

References

[1] P. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In B. Buxton and R. Cipolla, editors, *Computer Vision — ECCV '96*, pages 683–695, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[2] T. Brouver. F1 25 game 360° camera race start monaco gp! (new 360° feature gameplay). `https://www.youtube.com/watch?v=yZChkK0bKLc`, May 2025. YouTube video.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[4] Formula 1. Formula 1 official youtube channel. `https://www.youtube.com/user/Formula1`, 2025. Used for training footage and video material related to Formula 1 races and crashes.

[5] G. Hess, C. Lindström, M. Fatemi, C. Petersson, and L. Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving, 2025.

[6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.

[7] L. Pan, D. Barath, M. Pollefeys, and J. L. Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024.

[8] M. C. Parker and G. K. Hargrave. The development of a visualisation tool for acquired motorsport data. *Proceedings of the Institution of Mechanical Engineers, Part P*, 230(4):225–235, 2016.

[9] P. Schaefer. Fast-f1, 2019.

[10] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[11] D. Shi, S. Cao, L. Fan, B. Wu, J. Guo, R. Chen, L. Liu, and J. Ye. Trackgs: Optimizing colmap-free 3d gaussian splatting with global track constraints, 2025.

[12] M. Stoll, R. Krüger, T. Ertl, and A. Bruhn. Racecar tracking and its visualization using sparse data. 2013.

[13] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[14] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes, 2024.