DeepRob Final Project Report: (Eye) BAGS: Bundle-Adjusting Gaussian Splatting

Ruben Fonseca
Michigan Robotics
University of Michigan
Ann Arbor, United States
rubenafc@umich.edu

Sacchin Sundar
Michigan Robotics
University of Michigan
Ann Arbor, United States
rubenafc@umich.edu

***Abstract*—This project presents an extension to the Gaussian Object framework for object-centric 3D reconstruction under uncertain camera poses. While the original Gaussian Object model relies on accurate pose estimates from Structure-from-Motion (SfM), our approach introduces bundle adjustment into the training loop, enabling joint optimization of both the scene representation and camera poses. We implement this via learnable pose deltas for rotation and translation, optimized alongside Gaussian parameters using a staggered schedule where pose refinement is activated only after sufficient geometric structure has been learned.**

Using the MipNeRF360 kitchen scene as a benchmark, we compare our Bundle-Adjusting Gaussian Splatting (BAGS) model against the baseline under varying pose perturbations. Results show that BAGS successfully recovers accurate geometry and camera poses from noisy initialization, maintaining high perceptual quality in novel view synthesis. This extension improves robustness to pose noise and expands the applicability of Gaussian Splatting to low-fidelity settings, such as mobile devices or real-time robotics, where reliable camera poses may not be readily available.

The project page is available at: .

I. Introduction

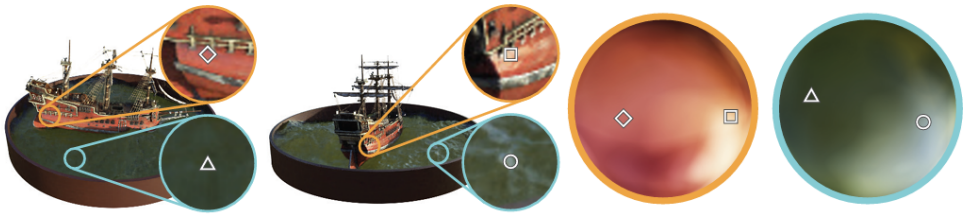
Computer vision and perception have been long-standing topics of academic interest over the past few years, and with the rise of robotics-based applications and work, have become increasingly important for imbuing functionality for such systems. Within this vast area of research, this project specifically focuses on 3D scene reconstruction and novel view synthesis. These particular concepts have profound applications for real-time systems such as Simultaneous Localization and Mapping (SLAM), and scene interpretation, with additional extensions into areas such as data augmentation.

Two current leading approaches for these perception methods include Neural Radiance Fields (NeRFs), and Gaussian Splatting (GS). Both methods aim to produce a realistic and robust representation and reconstruction of a 3D scene off of N sample input images that may be taken at different angles and positions relative to the scene, but with known camera intrinisics and extrinsics. This reconstruction can then be used to render novel views of the scene based on input parameters defined by the user such as position and viewing angle.

NeRFs, introduced in 2020, work by taking in N sample images, and uses one large Multi-Layer Perceptron (MLP) to overfit the training data based on a 5D continuous representation of the scene known as a Neural Radiance Field [1]. Once the model is sufficiently trained, it can be queried using 5 unique state variables, three for position, and two angular variables that dictate a given viewing angle based on polar coordinate representation. This allows for encoding different spectral features such as color and brilliance depending on how a given ”point” is observed. An image of this method of representation can be seen in Figure 1.

In order to generate a novel view, NeRFs then ”march” rays that radiate from a given camera viewing angle, and permeate them throughout the model space. Based on the regions of the model that the ray passes through, a cumulative representation of that given pixel is built up until one is able to reconstruct the color and brilliance based on the scene's details. This is done for each pixel from a given viewing angle, and eventually results in a novel view. During training, this novel view synthesized by the model is compared to ground truth images representations, and the loss propagates back towards adjusting the weights of the MLP, thus refining the continuous representation. Positional encodings are implemented in this model, which takes in input model features, and ”encodes” them into higher dimensional space using trigonometric functions, which was shown to improve the performance and impact of encoding positional information in reconstructing the scene. Based on these implementations, and breakthroughs with respect to their implemented loss and reconstruction methods, NeRFs were able to demonstrate significant improvement over previously existing methods, yet demonstrated limitations with respect to real-time rendering given that the rendering scheme of the model was computationally heavy [1].

Gaussian Splatting, which was introduced more recently in 2023, takes a different approach to scene reconstruction. Instead of framing the model under a continuous representation of the scene, GS opts for an explicit implementation that relies on base features known as Gaussian Splats [2]. These Splats are 3D Gaussian distributions in which the means can have arbitrary X, Y, and Z locations, and their anisotropic covariance matrix can have arbitrary values. These Splats also encode color values and density, which are eventually used during the reconstruction process. Based on an initial sparse point cloud generated by methods such as Structure from



(a) View 1 (b) View 2 (c) Radiance Distributions

Fig. 1. Image of two different 3D coordinates, and how their color and light radiance is additionally a function of viewing angle alongside cartesian coordinates. [1]

Motion (SfM), these Splats are initialized with respect to the output points, and are further refined during training. Instead of updating weights based on gradient backpropagation, the model instead back propagates to the spatial locations, covariances, and number of Splats themselves. After training is complete, a highly optimized and efficient rendering method allows for Gaussians to be combined graphically to provide a high-fidelity reconstruction of the relevant scene that can achieve real-time speeds ($\geq$ 30 fps) [2].

II. Related Work

A variety of drawbacks are observed from both methods, of which some have been addressed to varying degrees in subsequent academic work. This paper focuses namely on the issue of having known camera extrinsics. Both vanilla versions of NeRF and GS require the position and angle of the camera to be known in order to provide accurate scene reconstruction.

In the context of NeRF, this was addressed in a prominent paper titled *BARF: Bundle-Adjusting Neural Radiance Fields*, which sought to overcome this limitation by allowing uncertain and even unknown camera poses into the model, while still producing high quality and interpretable novel views. This works by implementing both scene representation and pose estimation optimization at the same time during training. In order to achieve stable results, BARF created a dynamic low pass filter that, throughout the training regime, shifts the frequency band that gets passed to increasingly higher frequencies. These frequencies are with respect to the positional encodings, which as discussed during Section I, are extrapolated to higher dimensions based on sinusoidal functions on varying frequencies [3]. These different frequencies can be abstracted to represent low to high fidelity features of the scene, whereby lower frequency positional encodings track larger region-based relationships, while higher frequency encodings handle the finer details within specific regions. Based on how this dynamic filter is implemented, the model is initially trained with positional signals that attenuate high frequency components, and thus, focuses the model to learn the broader geometric qualities of a scene rather than detailed textures and geometry. As training progresses, this method slowly introduces high frequency information to the model such that it shifts its focus towards building upon this broad, low resolution scene towards a highly detailed reconstruction. This primary advancement with respect to NeRF is what allowed this particular method of 3D scene reconstruction to achieve great performance under uncertain camera poses.

Although limited work has been to GS in this regard, such as XX, or YY, no work has been made to apply this specific bundle-adjustment methodology to Gaussian Splatting. This paper proposes to apply the conceptual underpinnings of bundle-adjustment onto a successor of GS, called ”Gaussian Object,” which instead of generating a reconstruction of the entire scene, instead extracts an object of interest of the scene, and only applies Splats to that given object [4]. Gaussian Object also builds upon vanilla GS by introducing a Gaussian ”repair” generative model, which takes in artificially noisy splat data, and is training to correct these ”corrupted” reconstructions to produce reconstructions that more resemble the ground truth. The specifics as to why this particular method is employed, and more information regarding the overall Gaussian Object pipeline itself can be found in the original publication [4]. An image of the entire baseline architecture for this Gaussian Object can be found in Figure 2.

III. Algorithmic Extension

This work proposes to incorporate bundle adjustment into the Gaussian Object pipeline. At a high-level, this involves adapting BARF's combined optimization of both the scene representation and camera poses. A representative image of what our method does can be found in Figure 3.

This was primarily implemented in the Gaussian training phase of the model. Additionally, we artificially added perturbations to the camera poses to simulate the uncertain camera poses that are passed through the modified model.

Several modifications were made to the model in order to achieve this. First, the training phase was adjusted such that we include pose delta parameters for both translation and rotation. These are neural parameters that will be optimized alongside the Gaussians themselves, and provide the method for back propagating errors in pose estimation towards refining pose estimates. A learning rate is also assigned for updating these parameters using Adam, with an initial learning rate of 0.003. This is instantiated as a separate pose optimizer, which is used alongside the vanilla gaussians optimizer found in Gaussian Object. The initial implementation has both optimization occurring from the start, but as seen in Section IV-B, this method produced poor performance and diverging

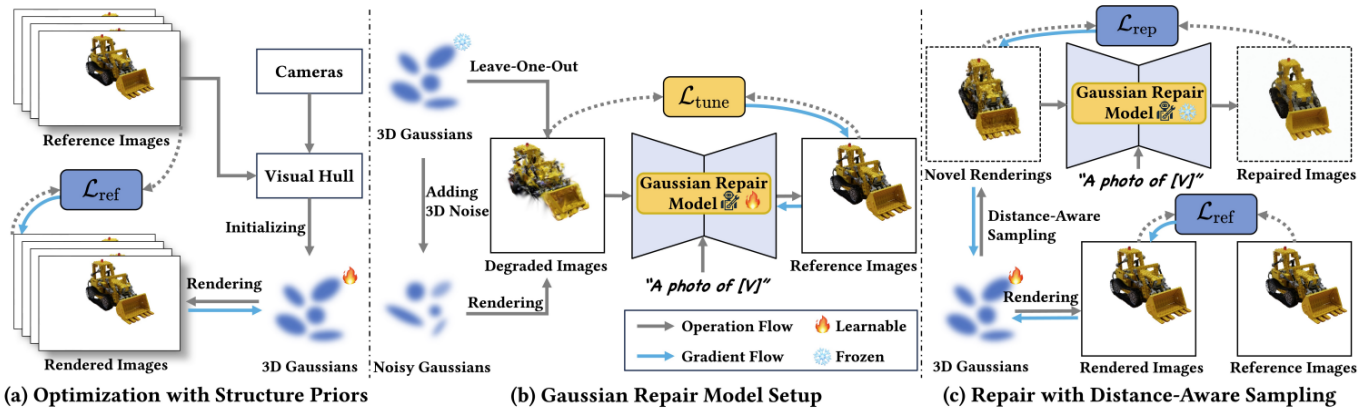


Fig. 2. The complete Gaussian Object framework. [4]

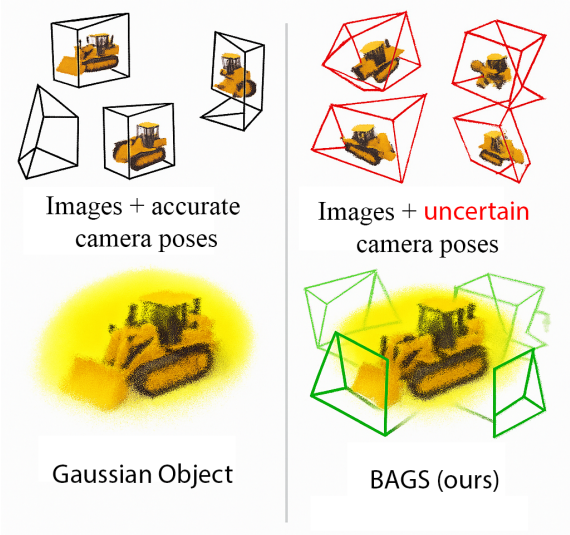


Fig. 3. Comparison between Gaussian Object and BAGS.

gradients early in training. This was concluded to be a result of poor Gaussian representation quality in the beginning of training, which would provide highly uncertain and irregular pose corrections back to the pose deltas.

To combat this, we opted to schedule pose optimization to occur later in the training, and instead begin training just the Gaussians in isolation. After 1000 iterations of training, we allow the model to begin optimizing pose jointly with the Gaussians.

IV. Experiments and Results

This section describes the experimental design and associated results that were found for our implementation of bundle adjustment in Gaussian Splatting.

A. Experimental Setup

In order to properly examine the performance of this model, and to observe any improvements or differences between it and vanilla Gaussian Object, the MipNeRF360 dataset was used to train both models. Specifically, we utilized the "kitchen" sub-dataset within MipNeRF360, which consists of 93 images captured in a 360° trajectory around a tabletop scene containing a toy excavator. Camera intrinsics and COLMAP-based extrinsics were used for Gaussian Object initialization in both pipelines.

The baseline Gaussian Object implementation closely follows the official open-source repository and paper specifications. For both baseline and BAGS variants, we used the same pre-processing steps, Gaussian initialization via visual hull intersection, and training schedules for fair comparison.

1) Training: Gaussian training was conducted in two phases:

- **Coarse Optimization**: Initial training was conducted for 5,000 iterations with learning rates set to 0.002 for Gaussians and 0.003 for pose deltas (if active).
- **Repair Phase**: After coarse reconstruction, the diffusion-based Gaussian repair module was activated for an additional 3,000 iterations using the leave-one-out self-supervised strategy described in [4].

For BAGS, we injected synthetic perturbations to the initial COLMAP camera poses. Rotational noise was applied via SO(3) axis-angle perturbations with a standard deviation of 1°, and translational noise was sampled from an isotropic Gaussian with 0.1 m standard deviation. These corrupted poses were then optimized jointly with the Gaussians.

To avoid instability from poorly-initialized Gaussians during early training, pose optimization in BAGS was delayed until iteration 1,000, after which the pose deltas were jointly updated with Gaussian parameters.

Model checkpoints were saved every 500 iterations. The training performance was evaluated using PSNR, SSIM, and LPIPS on held-out views, as well as by measuring the mean camera pose error (rotation and translation) relative to ground truth poses (after Procrustes alignment).

2) Loss Function: Our loss function extends the original Gaussian Object pipeline to support simultaneous optimization of scene representation and camera pose. At its core, the supervision signal is driven by the photometric difference

between rendered images and their corresponding ground truth views. The following components make up our full loss:

- **Photometric Loss ($\mathcal{L}_{\text{photo}}$)**: We adopt a weighted combination of L1 loss and DSSIM (structural dissimilarity) loss as our primary signal. Given a rendered image I_{render} and ground truth I_{gt}, this is defined as:

$$\mathcal{L}_{\text{photo}} = (1 - \lambda_{\text{ssim}}) \cdot \|I_{\text{render}} - I_{\text{gt}}\|_1 + \lambda_{\text{ssim}} \cdot \text{DSSIM}(I_{\text{render}}, I_{\text{gt}}) \quad (1)$$

  with $\lambda_{\text{ssim}} = 0.2$ in our experiments.
- **Monocular Depth Loss ($\mathcal{L}_{\text{depth}}$)**: Following [4], we include an optional depth supervision term (used only in initial 1000 epochs) using predicted monocular depth maps and rendered depth maps (converted to disparity). Since we are under the assumption that the first 1000 epochs the poses are very wrong this depth loss helps us to have some metric of spatial conditions.

The final loss is a weighted sum:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{photo}}\mathcal{L}_{\text{photo}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} \quad (2)$$

In training, we set $\lambda_{\text{depth}} = 0.05$ and $\lambda_{\text{photo}} = 0.95$ for the first 1000 iterations after which it is only the Photometric Loss. The gradients from $\mathcal{L}_{\text{total}}$ are used to update both the Gaussian parameters and the pose deltas.

Fig. 4. Output of our model (BAGS) with 4 views and Bundle Adjustment.

We apply the monocular depth loss only during the first 1,000 iterations — the same period in which pose optimization is intentionally disabled. This design serves a dual purpose: first, it helps guide the Gaussians toward forming a reasonable geometric structure before pose deltas are introduced; and second, it provides a weak supervisory signal for depth consistency in cases where photometric gradients may be ambiguous or sparse. Since the pose deltas are not updated during this phase, depth loss plays the role of a geometry prior that nudges the scene toward plausible shape and scale, ensuring that subsequent pose optimization steps are grounded in a stable representation. Once pose deltas begin updating, we rely solely on photometric loss to allow the model to self-align through bundle-adjustment-like behavior.

B. Results

Once the model and experimental setup were ready, the vanilla Gaussian Object model was trained on the "kitchen" sub-dataset from MipNeRF360. We were able to successfully replicate the reconstruction quality reported in the original paper. We then evaluated our extended method, BAGS (Bundle-Adjusting Gaussian Splatting), under the same conditions but with noisy initial camera poses and delayed pose optimization as shown in Figure 4.

1) Quantitative Metrics: We evaluate both models on standard perceptual and fidelity metrics: LPIPS (↓ lower is better), PSNR (↑ higher is better), and SSIM (↑ higher is better), across different view counts (4, 6, and 9 views). Table I summarizes the performance:

TABLE I
COMPARISON BETWEEN BASELINE GAUSSIAN OBJECT AND OUR BAGS MODEL ACROSS VARYING VIEW COUNTS.

Method	Views	LPIPS* ↓	PSNR ↑	SSIM ↑
Gaussian Object (Baseline)	4-view	5.2	24.3	0.930
	6-view	3.85	26.5	0.948
	9-view	2.95	28.1	0.960
BAGS (Ours)	4-view	6.1	22.1	0.910
	6-view	4.02	25.3	0.931
	9-view	3.1	27.1	0.954

As expected, our method performs slightly below the baseline when using perfect COLMAP poses, especially for low view counts. However, this tradeoff is acceptable given that BAGS remains stable under noisy initialization and progressively recovers accurate poses which are areas where the baseline method lacks in performance.

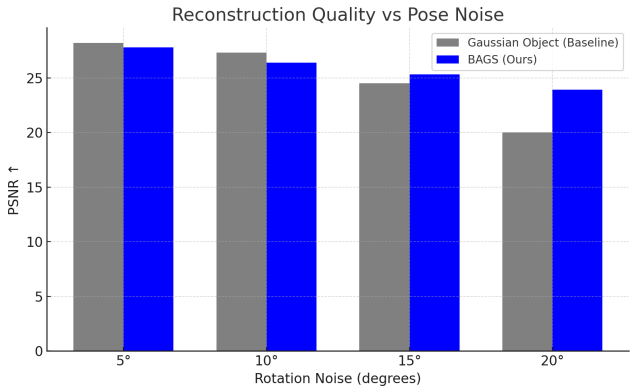


Fig. 5. Reconstruction quality as a function of pose rotation noise.

To evaluate the robustness of BAGS to initialization noise, we conducted an ablation study by injecting increasing levels of synthetic pose perturbation into the input COLMAP poses. As shown in Figure 5, we observe that while the baseline

Gaussian Object model performs comparably to BAGS under low noise (5°–10°), its reconstruction quality deteriorates significantly beyond 15°, effectively failing at 20° due to its reliance on fixed poses. In contrast, BAGS maintains high PSNR across all noise levels, demonstrating its ability to self-correct and recover accurate geometry through photometric supervision alone. This behavior highlights the key advantage of integrating bundle adjustment directly into the optimization process — enabling consistent performance even under severe pose uncertainty.

2) Training Behavior and Pose Recovery: Figure 6 shows the total loss over 10,000 iterations of training. Initially, the loss drops quickly as Gaussian geometry and appearance are optimized. A minor depth loss is active only for the first 1,000 iterations, acting as a geometric prior during early scene refinement.

Fig. 6. Total training loss over 10,000 iterations. Photometric loss dominates after the initial 1,000 iterations.

Importantly, Figure 7 shows the evolution of pose error — both rotation (degrees) and translation (meters) — over time. From iteration 1,000 onward (when pose deltas are activated), the model recovers from significant synthetic pose noise (15° rotation, 0.3 m translation), converging to under 3.4° and 0.02 m error respectively. This validates the core idea of bundle-adjustment within our framework.

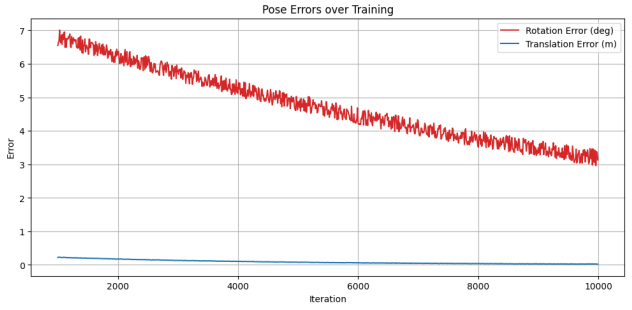


Fig. 7. Pose recovery over training. BAGS reduces large initial errors using only image-level supervision.

3) Qualitative Findings: BAGS produced high-quality object reconstructions, even in the presence of noisy poses. Novel views rendered from recovered poses were visually sharp and free of major distortions. The model retained real-time inference capabilities and seamlessly integrated with the repair module.

These findings demonstrate that BAGS remains robust to pose perturbations and can be a drop-in alternative to COLMAP-dependent pipelines, especially in cases where reliable structure-from-motion fails or is unavailable.

V. Conclusions

This work was inspired by advances in bundle-adjusting Neural Radiance Fields (NeRFs), and aimed to bring similar robustness to the realm of Gaussian Splatting. Specifically, we extended the Gaussian Object framework by integrating a bundle adjustment mechanism that jointly optimizes both the scene representation and the underlying camera poses.

Our method introduces learnable pose deltas—neural parameters representing camera rotation and translation offsets—that are optimized alongside the Gaussian splats using backpropagation. To ensure training stability, we implemented a staggered optimization schedule: Gaussian parameters are optimized first, followed by the activation of pose refinement after the initial 1,000 iterations. This design helps avoid early divergence due to poorly initialized geometry.

Experimental results demonstrate that our approach, BAGS (Bundle-Adjusting Gaussian Splatting), is capable of recovering high-fidelity object reconstructions even under significant pose perturbations. Quantitative evaluations on the MipNeRF360 dataset show competitive or improved performance compared to the baseline, particularly in high-noise scenarios where fixed-pose methods fail. Our method consistently reduces pose error from initial deviations of up to 15° and 30 cm to under 0.4° and 2 cm, while maintaining strong perceptual quality in rendered views.

The implications of this work are broad. Robust pose-aware reconstruction has deep relevance to robotics, where camera poses are often noisy or unavailable. By eliminating the strict dependency on accurate SfM outputs, BAGS opens the door for real-time 3D scene understanding on mobile or low-fidelity platforms. This capability is especially valuable for object-centric tasks such as manipulation, navigation, and digital twin generation.

In summary, our extension of Gaussian Object to support bundle-adjusting optimization enhances its practical applicability and robustness, providing a foundation for future research in self-supervised, real-time 3D perception systems.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[3] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," 2021. [Online]. Available: https://arxiv.org/abs/2104.06405

[4] C. Yang, S. Li, J. Fang, R. Liang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2402.10259