LoGS: Visual Localization via Gaussian Splatting with Fewer Training Images

Yuzhou Cheng[1], Jianhao Jiao[1*], Yue Wang[2], and Dimitrios Kanoulas[1, 3]

Abstract— **Visual localization involves estimating a query image's 6-DoF (degrees of freedom) camera pose, which is a fundamental component in various computer vision and robotic tasks. This paper presents LoGS, a vision-based localization pipeline utilizing the 3D Gaussian Splatting (GS) technique as scene representation. This novel representation allows high-quality novel view synthesis. During the mapping phase, structure-from-motion (SfM) is applied first, followed by the generation of a GS map. During localization, the initial position is obtained through image retrieval, local feature matching coupled with a PnP solver, and then a high-precision pose is achieved through the analysis-by-synthesis manner on the GS map. Experimental results on four large-scale datasets demonstrate the proposed approach's SoTA accuracy in estimating camera poses and robustness under challenging few-shot conditions.**

I. INTRODUCTION

A. Motivation

In an increasingly automated world, the ability of robots to understand and navigate their surrounding environment has become crucial for numerous applications, ranging from autonomous vehicles and extended reality (XR) to industrial automation and disaster response. Visual localization is at the core of capabilities, allowing robots to accurately determine their six degrees of freedom (6-DoF) position and orientation.

Current visual localization methods fall into three major types: absolute pose regression (APR) [1]–[6], structure-based [7]–[15], and analysis-by-synthesis [16]–[22] methods. **APR** estimates the camera pose directly from neural networks but need help with accuracy and generalization. **Structure-based** approaches contain feature matching-based (FM) [7]–[9] and scene coordinate regression (SCR) [10]–[15]. FM identifies 2D-3D correspondences between image projections and spatial coordinates in the point cloud, while SCR directly predicts such correspondences each pixel through a trained neural network. Typically, a geometric solver such as the PnP-RANSAC estimates the camera poses these 2D-3D correspondences. FM pipelines have been

[1]Robot Perception and Learning Lab, Intelligent Robotics, Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK. {yuzhou.cheng.23, ucacjji, d.kanoulas}@ucl.ac.uk
[2]Zhejiang University, Hangzhou, Zhejiang, China. wangyue@iipc.zju.edu.cn
[3]AI Centre, Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK and Archimedes/Athena RC, Greece.

*Corresponding Author: Jianhao Jiao

widely adopted, but their accuracy is usually lower than that of SCR if the model is trained with sufficient data. Nevertheless, many SCR networks are specifically designed for localization, making them an additional burden for robots.

Recently, iNeRF [16], [17] emerged as an **analysis-by-synthesis** approach that iteratively inverts neural radiance fields (NeRFs) to align camera poses. Nonetheless, these approaches suffer from time limitations due to low rendering speed. 3D Gaussian Splatting (GS) [23], a paradigm-shifting Novel View Synthesis technique, achieves comparable render quality and real-time rendering. It rasterizes a collection of Gaussian ellipsoids to approximate a scene's appearance. Analysis-by-synthesis localization using 3DGS as the map representation [21], [22] has started to gain attention. They have yet to be tested on large-scale datasets [1], [10], [24] and lack comparisons to baselines in other categories.

B. Contributions

This paper introduces a novel visual localization pipeline, termed LoGS, which employs GS as the foundational map structure. Especially, LoGS addresses challenges related to data scalability. As we want: "You don't need a lot to make a difference." Training a environmental representation with only dozens or even just a few images generally alleviates data scarcity and reduces resource requirements, but at the cost of accuracy decay [25]. This few-shot setting [25] tests a pipeline's robustness and generality as well, where many of the aforementioned neural network-based methods tend to fail. Our method, on the contrary, outperforms the state-of-the-art (SoTA) using only 0.5% to 1% of the training images. For example, by utilizing only 20 out of 4000 images, we achieve a median translation error of 0.5 cm and a median rotation error of $0.16°$ (see TABLE II) in the CHESS scene from the 7-scenes dataset [10]. This is crucial for practical applications that require rapid deployment.

We obtain a point cloud for GS map initialization by performing Structure-from-Motion (SfM) with advanced feature-matching. Then, we utilize depth clues and regularization strategies to build a high-resolution GS map. LoGS estimates a rough pose through PnP-RANSAC on the SfM point cloud when localization starts. LoGS then minimizes the photometric loss between the query image and the rendered images on the GS map to obtain an exceptionally accurate final pose. We also propose masking policies to choose the most representative pixels for residual comparison. Our pipeline achieves SoTA accuracy across four large-scale localization benchmarks [1], [10], [26], [27]

covering indoor and outdoor environments. In summary, the contributions of this work are threefold:

- We present a novel visual localization pipeline with 3DGS as the core map representation, which operates in a hierarchical manner.
- Extensive experiments have been conducted on four real-world full/few-shot benchmarks. LoGS is on par with or sets new baselines for these datasets.
- We demonstrate the practical effectiveness of adding depth clues and regularization strategies for GS map formation and the usefulness of adding different masks for photometric residuals' comparison.

II. RELATED WORK

A. Absolute Pose Regression

Absolute Pose Regression (APR) entails training neural networks to directly regress a 6-DoF camera pose from an image. PoseNet [1] marks the first APR method utilizing a CNN framework. Following PoseNet, enhancements such as temporal information [2], geometric loss function [3], and photometric consistency [4] have refined the accuracy of APR. Applying Transformer mechanisms in Multi-Scene APR [5] and map-relative pose regression [6] have also propelled the field. However, APR still suffers from precision and generality [28], [29]. Our LoGS overcomes the significant drop in accuracy prediction experienced by these direct methods on few-shot training images.

B. Structure-based Localization

Structure-based localization involves: 1) identifying correspondences between 2D image pixels and 3D scene points and 2) solving for the camera pose through a geometric solver such as PnP-RANSAC. Traditional FM approaches [7]–[9] establish correspondences via 2D-2D feature matching while recent Scene Coordinate Regression (SCR) methods regress pixels to 3D coordinates. Pioneering SCR uses Regression Forests [10] for RGB-D camera localization. Neural network approaches have gradually outperformed Regression Forests in recent years. DSAC [11] [12] employs CNNs to predict scene coordinates and score hypotheses, introducing a differentiable RANSAC algorithm. Region classification [13] and the segmentation branch [14] are later introduced to enhance scene understanding. There is also attention on scene-agnostic coordinates regression [15] where model parameters and scenes are independent. Extending FM's pipeline, LoGS further improves the accuracy and achieves SoTA results on the 7-scenes and Cambridge Landmarks datasets through an additional refinement step.

C. Analysis-by-synthesis

Analysis-by-synthesis methods optimize the camera pose by reducing the L_1 or L_2 norm of the difference in pixel-level features between a synthesized image and the query image. They either independently achieve relocalization or serve as pose refinement modules. This approach has its roots in many visual tracking components [30]–[32]. These works [16], [17], [20] use iNeRF to refine camera poses through photometric loss while [18], [19] render and align higher dimensional features for each pixel. Recent analysis-by-synthesis methods use 3D GS as the scene representation, as seen in works [21] [22]. Nevertheless, pose refinements of many methods mentioned above are prone to converge to local optima. To tackle this issue, LoGS builds a fine-grained GS map with depth clues and regularization strategies and designs masks that filter pixels to avoid local convergence.

III. PRELIMINARIES

The 3D GS [23] is built upon three components [33]. The first component is the basic scene representation. 3D Gaussians in space are colored ellipsoids whose transparency gradually decays from its center point according to a Gaussian distribution. The second component focuses on optimizing the properties of the 3D Gaussians. During optimization, 3DGS adopts an adaptive density control—adds and removes 3D ellipsoids to produce a compact and unstructured scene representation, typically resulting in several million Gaussians for a target scene. The last element of 3DGS is a rapid rendering strategy that leverages tile-based rasterization.

We define a 3D Gaussian by its opacity α, color c, center position $\boldsymbol{\mu}$, and 3D covariance matrix $\boldsymbol{\Sigma}$:

$$G(\mathbf{x}) = [\alpha, c]e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \qquad (1)$$

3D Gaussians are projected into the 2D image plane for rendering. The resulting 2D covariance matrix $\boldsymbol{\Sigma}'$ is derived from the viewing transformation $\mathbf{W}$ and the 3D covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}' = \mathbf{J}\mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^\top\mathbf{J}^\top. \qquad (2)$$

For a given pixel position $\mathbf{p}$, the distances to all overlapping Gaussians generate a sorted list of Gaussians $\mathcal{N}$ [33]. Alpha compositing is applied to determine the pixel's color:

$$C(\mathbf{p}) = \sum_{i=1}^{|\mathcal{N}|} c_i \alpha_i' \prod_{j=1}^{i-1}(1 - \alpha_j'), \qquad (3)$$

where c_i is the color after training. The opacity α_i' is the multiplication outcome of the trained opacity α_i and projected position within the Gaussian: $\alpha_i' = \alpha_i \cdot \exp\left(-\frac{1}{2}(\mathbf{x}' - \boldsymbol{\mu}_i')^\top \boldsymbol{\Sigma}_i'^{-1}(\mathbf{x}' - \boldsymbol{\mu}_i')\right)$, where $\mathbf{x}'$ and $\boldsymbol{\mu}_n'$ are coordinates in the projected space. By accumulating the distance along the ray, we can also define a differentiable rendered depth:

$$D(\mathbf{p}) = \sum_{i=1}^{|\mathcal{N}|} d_i \alpha_i' \prod_{j=1}^{i-1}(1 - \alpha_i'), \qquad (4)$$

which can be compared with the input depth map if the query image has a depth channel. Additionally, we render an occupancy image to determine visibility:

$$O(\mathbf{p}) = \sum_{i=1}^{|\mathcal{N}|} \alpha_i' \prod_{j=1}^{i-1}(1 - \alpha_i'). \qquad (5)$$

This occupancy image measures the confidence level of the Gaussian ellipsoids' contribution at a given pixel.
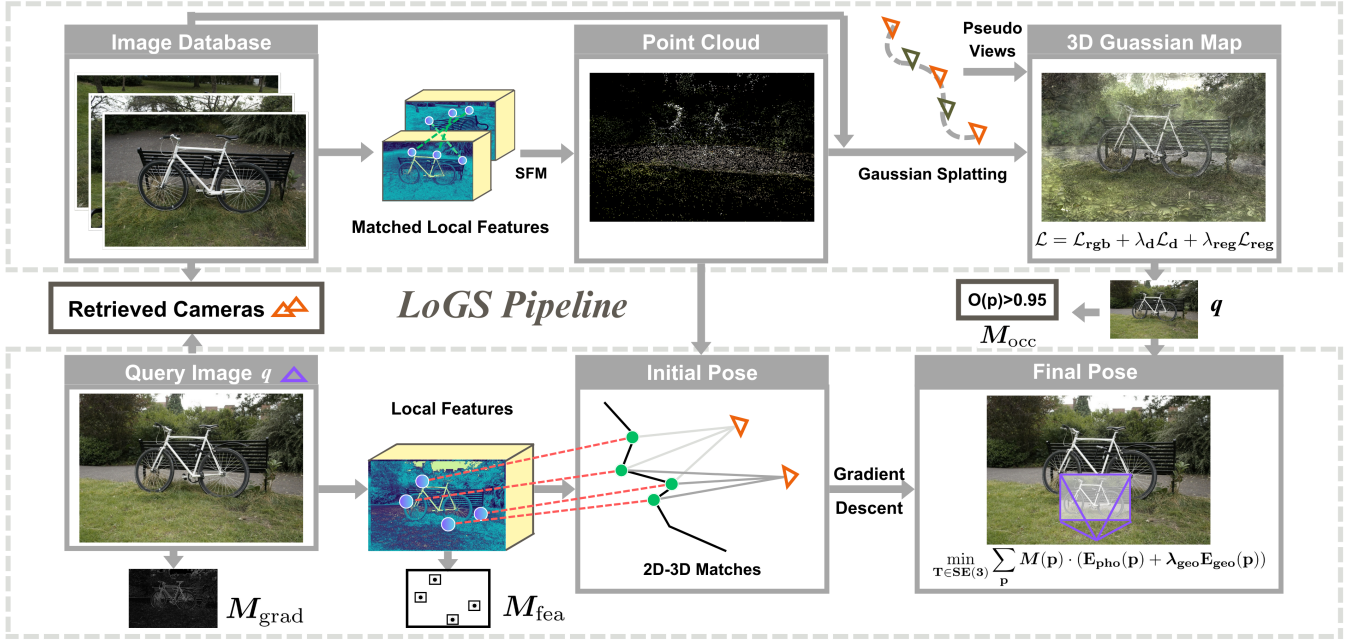
Fig. 1: An illustration of the LoGS pipeline where the localization process aligns with the mapping.

IV. METHODOLOGY

A. Mapping

SfM: A random initial distribution can result in some Gaussians being unable to be optimized to their ideally optimal positions, leading to artifacts such as floaters. This, in turn, can affect the final image rendering quality. If we already have an SfM point cloud distribution at the start of GS map construction, we can initialize an ellipsoid at each point, which gives a relatively good representation from the beginning. Thus, LoGS first utilizes SuperPoint [34] and SuperGlue [8] to extract features and perform feature matching on the images in the database. Then, an accurate sparse point cloud is constructed through SfM triangulation.

GS map: Given all the renders, we design a loss function to optimize learnable parameters in the GS map. We first reduce the photometric residual:

$$\mathcal{L}_{rgb} = \left\| C - \bar{C} \right\|_1, \tag{6}$$

where C is the rendered color image from the Gaussians and ground truth pose T, and $\bar{C}$ is the ground truth color image.

When the images for training have a depth channel, we similarly express the geometric loss using the L_1 norm:

$$\mathcal{L}_d = \left\| D - \bar{D} \right\|_1, \tag{7}$$

where D is the rendered depth image and $\bar{D}$ is the ground truth depth image.

When ground truth depth is absent, we generate monocular depth maps $\hat{D}$ for training images using the pre-trained Dense Prediction Transformer (DPT) [35] to regularize the training. We apply a relaxed relative loss using Pearson correlation [36] between the estimated depth $\hat{D}$ and rendered depth D. This method measures the distributional differences between the two depth maps:

$$\mathcal{L}_{reg}(D, \hat{D}) = \frac{\text{Cov}(D, \hat{D})}{\sqrt{\text{Var}(D)\text{Var}(\hat{D})}}. \tag{8}$$

Experimental results show that adding this regularization term improves the quality of novel view synthesis even when the dataset includes ground truth RGB-D images. This improvement is due to the continuity of pseudo depths, which filter out isolated artifact Gaussians.

In sum, we reach the following optimization objective for each training image in the image database:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_d \mathcal{L}_d + \lambda_{reg} \mathcal{L}_{reg}. \tag{9}$$

When there are very few training images, the coverage of the scene is incomplete, and over-fitting happens. LoGS applies the $\mathcal{L}_{reg}$ loss on pseudo-views. Given a set of N train-view poses $\{T_1, \ldots, T_N\}$, where each pose $T_i = \{R_i, t_i\}$, we find a permutation π that minimizes the total pairwise distance:

$$\min_{\pi} \sum_{i=1}^{N-1} d(T_{\pi(i)}, T_{\pi(i+1)}), \tag{10}$$

where $d(T_i, T_j) = \|t_i - t_j\|_2$ is the L_2 translation error.

K pseudo views are interpolated between consecutive poses $T_{\pi(i)}$ and $T_{\pi(i+1)}$ with Spherical Linear Interpolation (SLERP):

$$\begin{cases} t^{(k)} = (1 - \alpha(k))t_i + \alpha(k)t_{i+1} \\ R^{(k)} = \text{SLERP}(R_i, R_{i+1}, \alpha(k)) \end{cases}, \tag{11}$$

where $\alpha(k) = \frac{1 - \cos\left(\frac{k\pi}{K+1}\right)}{2}$. A series of smoothly transitioning pseudo views are thus generated between real views.

B. Localization

Initial Pose Estimation: To estimate an initial pose, we employ a feature matching-based approach [37] on the SfM point cloud, which consists of prior retrieval, covisibility clustering, local matching, and localization.

1) Prior Retrieval: Compare the query image with database images using global descriptors from NetVLAD [38]. The k-nearest neighbors represent potential locations within the map.

2) Covisibility Clustering: cluster the neighbors based on the covisibility of 3D structures—two frames belong to the same place if they observe common 3D points. Then, we perform independent local searches in each place.

3) Local Matching and Localization: Beginning with the place that contains the most number of nearest neighbors, we traverse through every place. We obtain the geometric relationship by matching local descriptors between the 3D points in the place and critical points in the query image with SuperGlue [8]. Finally, we check the geometry consistency and estimate a pose by solving the PnP-RANSAC problem.

Iterative Pose Refinement: Given limitations in matching accuracy, point cloud precision, and the visibility overlap of retrieved images, the initial pose is partially accurate. However, it is a strong starting point for further pose refinement. 3DGS enables a direct, nearly linear (projective) gradient flow between the parameters and the rendered output. As a result, we refine the pose through iterative updates using gradient-based optimization, taking advantage of differentiable rendering for both RGB and depth:

$$\hat{T} = \arg \min_{T \in \text{SE}(3)} \mathcal{L}(T \mid \mathcal{I}, \mathcal{G}), \tag{12}$$

where $\mathcal{I}$ is the query image and $\mathcal{G}$ is the GS map.

At each iteration, we optimize and update the camera pose with respect to the GS map. In the monocular case, we minimize the following photometric residual [39]:

$$E_{pho}(\mathbf{p}) = |(e^a \cdot C(\mathbf{p}) + b) - \bar{C}(\mathbf{p})|, \tag{13}$$

where $C(\mathbf{p})$ is the color of the rendered image at pixel $\mathbf{p}$, and $\bar{C}(\mathbf{p})$ is the color of the observed image at the same pixel position. We optimize affine brightness parameters a and b for varying exposure. These two parameters are vital for controlling illumination changes, especially in outdoor environments.

When a depth channel exits, we similarly define the depth residual between rendered depth and ground truth depth at a given pixel $\mathbf{p}$:

$$E_{geo}(\mathbf{p}) = |D(\mathbf{p}) - \bar{D}(\mathbf{p})|. \tag{14}$$

To mitigate the impact of noise in the scene representation that could distort the rendered images, we carefully designed a mask to select only information-rich pixels for comparison. This filter results in a more robust objective function and prevents the optimizer from sinking into local optima.

An edge detector is used to select pixels above a certain threshold, capturing important structural information in the image and reducing the amount of data that needs to be processed. The gradient mask M_{grad} is then defined as:

$$M_{\text{grad}}(\mathbf{p}) = \begin{cases} 1, & \text{if } |\nabla(\mathbf{p})| > \tau_{\text{grad}}, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

where $|\nabla(\mathbf{p})|$ is the gradient magnitude of the Scharr operator.

We used SuperPoint [34] during SfM to extract local descriptors and key points. These points effectively identify corners or blobs in the image. Around each significant feature point, we select a small area as the region of interest:

$$M_{\text{fea}}(x, y) = \begin{cases} 1, & \text{if } \exists(x_i, y_i) \ s.t. \ |x - x_i| \leq \tau_{\text{fea}} \\ & \text{and } |y - y_i| \leq \tau_{\text{fea}}, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Opacity mask M_{occ} focuses on pixels that contain Gaussian ellipsoid information rather than on arbitrary pixels:

$$M_{\text{occ}}(\mathbf{p}) = \begin{cases} 1, & \text{if } O(\mathbf{p}) > \tau_{\text{occ}}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

To summarize, we reach the following optimization objective for the pose:

$$\min_{T \in SE(3)} \sum_{\mathbf{p}} M(\mathbf{p}) \cdot (E_{pho}(\mathbf{p}) + \lambda_{geo} E_{geo}(\mathbf{p})), \tag{18}$$

where $M = (M_{\text{grad}} \cup M_{\text{fea}}) \cap M_{\text{occ}}$.

V. EXPERIMENTS

A. Datasets

We choose the Mip-NeRF 360 [26] and LLFF [27] datasets to compare analysis-by-synthesis baselines [16], [21]. The Mip-NeRF 360 dataset consists of nine scenes, five outdoors and four indoors, while the LLFF has complex real-world scenes for rendering novel views. To compare with other mainstream localization methods, we choose the widely-used indoor 7-scenes dataset [10] and the outdoor Cambridge Landmarks dataset [1].

B. Metrics

Translation error is the norm of the difference between the ground truth pose's position and the estimated pose's position, while the rotation error is the angle between the ground truth orientation and the estimated orientation. **Success rate** corresponds to the proportion of rotation error less than a threshold (5 degrees) and the proportion of translation error less than a threshold (5 cm) [11], [13], [25]. **Median pose error** refers to the median of the translation errors and the median of the rotation errors among all testing images.

C. Implementation Details

Each scene is iterated 30,000 times during GS map construction. Every 20 iterations, a pseudo view is randomly selected to add additional regularization. The weight λ_d of $\mathcal{L}_d$ is 0.05 and the weight λ_{reg} of $\mathcal{L}_{reg}$ is 0.01. For localization, we choose the Adam optimizer for gradient descent. The learning rates, including angular, translational,

TABLE I: Quantitative comparison of methods on the 7-Scenes dataset with DSLAM ground truth. Results: AS [7], HLoc [8], [9], HSCNet [13], DSAC* [12], SP+Reg [25], FSRC [25]. Fewshot results are from [25].

Methods (DSLAM GT)	#Images	Original training (median pose error in cm/°)						#Images	Few-shot training (median pose error in cm/°)					
		AS	HLoc	HSCNet	DSAC*	ACE	Ours		HLoc	DSAC*	HSCNet	SP+Reg	FSRC	Ours
CHESS	4000	3/0.87	2/0.85	2/0.7	2/1.10	2/0.7	2.0/0.62	20	4/1.42	3/1.16	4/1.42	4/1.28	4/1.23	3/1.00
FIRE	2000	2/1.01	2/0.94	2/0.9	2/1.24	2/0.9	1.8/0.70	10	4/1.72	5/1.86	5/1.67	5/1.95	4/1.53	2/0.90
HEADS	1000	1/0.82	1/0.75	1/0.9	1/1.82	1/0.6	1.0/0.64	10	4/1.59	4/2.71	3/1.76	3/2.05	2/1.56	2/0.99
OFFICE	6000	4/1.15	3/0.92	3/0.8	3/1.15	3/0.8	2.4/0.69	30	5/1.47	9/2.21	9/2.29	7/1.96	5/1.47	4/1.13
PUMPKIN	4000	7/1.69	5/1.30	4/1.0	4/1.34	4/1.1	4.0/1.03	20	8/1.70	7/1.68	8/1.96	7/1.77	7/1.75	7/1.85
REDKITCHEN	7000	5/1.72	4/1.40	4/1.2	4/1.68	4/1.3	3.4/1.13	35	7/1.89	7/2.02	10/2.63	8/2.19	6/1.93	5/1.64
STAIRS	2000	4/1.01	5/1.47	3/0.8	3/1.16	4/1.1	3.2/0.81	20	10/2.21	18/4.8	13/4.24	120/27.37	5/1.47	7/1.85

TABLE II: Quantitative comparison of methods on the 7-Scenes dataset with SfM ground truth. Results: MS-Transf [40], Marepo [6], DFNet [41], DSAC* [12], ACE [42], GLACE [43], MCLoc [18], NeFeS [19], NeRFMatch [20].

Methods (SfM GT)	#Images	Absolute pose regression			Scene coordinate regression			Analysis-by-synthesis				#Images	Ours
		MS-Transf	Marepo	DFNet	DSAC*	ACE	GLACE	MCLoc	NeFeS	NeRFMatch	Ours		
CHESS	4000	11/6.4	1.9/0.83	3/1.1	0.5/0.17	0.5/0.18	0.6/0.18	2/0.8	2/0.8	0.9/0.3	0.4/0.10	20	0.5/0.16
FIRE	2000	23/11.5	2.3/0.92	6/2.3	0.8/0.28	0.8/0.33	0.9/0.34	3/1.4	2/0.8	1.1/0.4	0.6/0.18	10	0.8/0.26
HEADS	1000	13/13.0	2.1/1.24	4/2.3	0.5/0.34	0.5/0.33	0.6/0.34	3/1.3	2/1.4	1.5/1.0	0.5/0.26	10	0.7/0.48
OFFICE	6000	18/8.1	2.9/0.93	6/1.5	1.2/0.34	1/0.29	1.1/0.29	4/1.3	2/0.6	3.0/0.8	0.7/0.22	30	1.2/0.34
PUMPKIN	4000	17/8.4	2.5/0.88	7/1.9	1.2/0.28	1.2/0.28	1/0.22	5/1.6	2/0.6	2.2/0.6	0.7/0.22	20	1.1/1.29
REDKITCHEN	7000	16/8.9	2.9/0.98	7/1.7	0.7/0.21	0.8/0.20	0.8/0.20	6/1.6	2/0.6	1.0/0.3	0.5/0.14	35	0.9/.022
STAIRS	2000	29/10.3	5.9/1.48	12/2.6	2.7/0.78	2.9/0.81	3.2/0.93	6/2.0	5/1.3	10.1/1.7	1.6/0.43	20	4.1/1.10

TABLE III: Quantitative comparison of methods on LLFF and Mip-NeRF 360.

Methods (<0.05 unit/<5°)	iNerf (δ_s)	iComMa (δ_s)	iComMa (δ_m)	Ours	Ours (few-shot)
LLFF	94.8/72.2	99.1/99.3	75.4/98.2	100/100	100/100
Mip-NeRF 360	85.6/79.6	86.7/90.6	68.8/74.8	100/100	94.7/99.9

and brightness parameters, are all set to 0.01. The three thresholds for M_{grad}, M_{fea}, and M_{occ} are set as 1, 10, and 0.99 respectively. The weight λ_{geo} for depth residual $E_{geo}(p)$ is set as 0.01. We train and evaluate all datasets on one RTX 4080 Ti GPU with a memory of 16GB.

D. Comparison

Mip-NeRF 360 and LLFF: TABLE III shows the success rates of iNeRF, iComMa, and LoGS in the LLFF and Mip-NeRF 360 datasets. iNeRF and iComMa depends heavily on pose initialization. δ_s corresponds to a minimal margin initialization where the translation is randomly set from $\pm[0, 0.1]$ in units and the rotation from $\pm[0, 20]$ in degrees. δ_m corresponds to a middle margin initialization where the translation is randomly set from $\pm[0.1, 0.2]$ and the rotation from $\pm[20, 40]$. We first follow the same split setting as iNeRF and iComMa, where most images are used for map construction while only five are used for localization. LoGS perfectly solved this localization problem when tested on five images, achieving a 100% recall rate with rotation errors less than 5 degrees and translation errors under 0.05 units.

Discovering this, we further explored a much more difficult few-shot setting, using the Mip-NeRF 360 dataset by uniformly selecting one-tenth of the data from each scene for training (from 12 to 31 images), with the remaining data reserved for testing. For the LLFF dataset, one-fifth of the data was used for training (from 4 to 12 images). Even with such scarce posed images, LoGS reaches higher success rates than the other two methods, demonstrating an advanced competence for accurate pose estimation. Our success on these two datasets is partially due to the new training loss, which significantly improves the rendering quality of the GS map.

7-scenes: Each cell of Table I contains the median translation error (in centimeter) and the median rotation error (in degree), respectively. The left side of the table shows the localization accuracy obtained by each approach being trained on the full training set, while the right side shows the accuracy of the few-shot training sets. As the training data decreases, the localization error increases for all methods. The ability to achieve accurate localization under such extreme conditions demonstrates the stability of a system.

With all the data, LoGS achieved the best results across seven scenes. When using only a handful of images, it outperforms other methods in multiple scenes, with the median rotational error in the PUMPKIN scene being nearly identical to the best result, while the translational and rotational errors in the STAIRS scene show a relative gap compared to FSRC [25]. Upon analysis, we believe this is due to 1) the similarly colored, repetitive structure of the multi-layered steps in the stairs and 2) the significant deviation in the initial pose estimation, which together cause the model to converge to a local optimum.

Fig. 2: Median error pose illustration (full-training on SfM ground truth). The bottom-left region of each plot is the original image. The upper-right part corresponds to the rendered image from Gaussian Splatting and the estimated pose. The first 7 plots are from the 7-scenes datasets and the last two are from the Cambridge Landmarks dataset.

TABLE IV: Quantitative comparison of methods on the Cambridge Landmarks dataset. SCRNet refers to [13].

Methods	# Images	Original training (median pose error in cm/°)						# Images	Few-shot training (median pose error in cm/°)					
		AS	HLoc	SCRNet	HSCNet	DSAC*	Ours		HLoc	DSAC*	HSCNet	SP+Reg	FSRC	Ours
GREATCOURT	1531	24/0.13	16/0.11	125/0.6	28/0.2	49/0.3	12.7/0.09	16	72/0.27	NA	NA	NA	81/0.47	68/0.20
KINGS-COLLEGE	1220	13/0.22	12/0.20	21/0.3	18/0.3	15/0.3	10.8/0.19	13	30/0.38	156/2.09	47/0.74	111/1.77	39/0.69	24/0.33
OLDHOSPITAL	895	20/0.36	15/0.30	21/0.3	19/0.3	21/0.4	14.6/0.31	9	28/ 0.42	135/2.21	34/0.41	116/2.55	38/0.54	28/0.43
SHOPFACADE	229	4/0.21	4/0.20	6/0.3	6/0.3	5/0.3	4.1/0.19	3	27/1.75	NA	22/1.27	NA	19/0.99	39/2.39
STMARYSCHURCH	1487	8/0.25	7/0.21	16/0.5	9/0.3	13/0.4	6.9/0.20	15	25/0.76	NA	292/8.89	NA	31/1.03	22/0.67

We also train on SfM ground truth and obtain the median error results for all 7 scenes (see TABLE II). Brachmann et al. [44] suggest no significant advantage of one ground truth over the other on the 7-Scenes dataset. However, different localization methods show varying accuracy depending on the type used. Moreover, NeRF-synthesis methods [19], [20] have demonstrated that rendered images tend to have higher quality when using SfM ground truth, and we observed the same phenomenon with the GS map. LoGS sets a new baseline for analysis-by-synthesis approaches trained with whole data. Utilizing only a few dozen images, We found that LoGS achieve median translation error around a centimeter (except the STAIRS scene). This is a remarkably impressive result, as the achieved accuracy is comparable to SCR methods trained with one hundred times more data.

Cambridge Landmarks: TABLE IV summarizes the median pose errors in centimeter and degree. LoGS, in general, demonstrates accuracy improvements over state-of-the-art feature matching-based methods on whole dataset training. We then test LoGS with around 1% data. NA indicates failure: median translation error greater than 500 centimeter. First, it is worth noting that many methods using neural networks as map frameworks, such as DSAC*, failed. This is because these methods employ complex network structures to enhance learning capability, which leads to poor generalization with a small training set. Nevertheless, we achieved the best accuracy in four scenes, setting a new

benchmark. Overall, LoGS demonstrated robustness in large-scale outdoor scenes with limited training data. Our "failure" on the SHOPFACADE scene is mainly because it is a corner, and three simple RGB images made it difficult for 3DGS to determine depth, resulting in a final map with a few overlapping shadows.

VI. CONCLUSION

This paper broadens the boundaries of mobile robotics [45]–[48] by exploring visual localization using 3DGS as a map representation. Scene Coordinate Regression and Absolute Pose Regression can accurately estimate poses with abundant posed images but tend to fail when training viewpoints are insufficient. In contrast, feature-based methods can predict poses under both conditions but with less accuracy. Our pipeline LoGS achieved high-precision image rendering from the GS map by optimizing the initial point cloud, loss function, and regularization methods. Based on that, LoGS combined multiple masks, selected the most representative pixels to compare photometric loss on RGB(D) channels, and utilized gradient descent to obtain an accurate pose from an initial estimation. Our method outperformed baselines in full/few-shot settings on four large-scale datasets, achieving leading-edge results. Future directions on finer GS reconstruction (e.g., illumination changes), new masking strategies, and GS map compression that reduces memory and increases localization speed can improve the work.

REFERENCES

[1] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[2] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6856–6864, 2017.

[3] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017.

[4] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, pages 1175–1185. IEEE, 2021.

[5] Yoli Shavit, Ron Ferens, and Yosi Keller. Multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021.

[6] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20665–20674, 2024.

[7] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

[8] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[9] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12716–12725, 2019.

[10] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[11] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017.

[12] Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021.

[13] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020.

[14] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. Vs-net: Voting with segmentation for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6101–6111, 2021.

[15] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 42–51, 2019.

[16] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

[17] Yunzhi Lin, Thomas Müller, Jonathan Tremblay, Bowen Wen, Stephen Tyree, Alex Evans, Patricio A Vela, and Stan Birchfield. Parallel inversion of neural radiance fields for robust pose estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9377–9384. IEEE, 2023.

[18] Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The unreasonable effectiveness of pre-trained features for camera pose refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12798, 2024.

[19] Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20987–20996, 2024.

[20] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. *arXiv preprint arXiv:2403.09577*, 2024.

[21] Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*, 2023.

[22] Matteo Bortolon, Theodore Tsesmelis, Stuart James, Fabio Poiesi, and Alessio Del Bue. 6dgs: 6d pose estimation from a single image and a 3d gaussian splatting model. In *ECCV*, 2024.

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[24] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332. IEEE, 2016.

[25] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *2022 International Conference on 3D Vision (3DV)*, pages 393–402. IEEE, 2022.

[26] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[27] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019.

[28] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3302–3312, 2019.

[29] Changkun Liu, Shuai Chen, Yukun Zhao, Huajian Huang, Victor Prisacariu, and Tristan Braud. Hr-apr: Apr-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation. *arXiv preprint arXiv:2402.14371*, 2024.

[30] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011.

[31] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision (ECCV)*, pages 834–849. Springer, 2014.

[32] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[33] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.

[34] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[35] Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021.

[36] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00451*, 2023.

[37] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, pages 456–465. PMLR, 2018.

[38] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[39] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.

[40] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021.

[41] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.

[42] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053, 2023.

[43] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21562–21571, 2024.

[44] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6218–6228, 2021.

[45] Dimitrios Kanoulas, Nikos G Tsagarakis, and Marsette Vona. Curved patch mapping and tracking for irregular terrain modeling: Application to bipedal robot foot placement. *Robotics and Autonomous Systems*, 2019.

[46] Dimitrios Kanoulas, Nikos G. Tsagarakis, and Marsette Vona. rxk-infu: Moving volume kinectfusion for 3d perception and robotics. In *18th IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2018.

[47] Jianhao Jiao, Ruoyu Geng, Yuanhang Li, Ren Xin, Bowen Yang, Jin Wu, Lujia Wang, Ming Liu, Rui Fan, and Dimitrios Kanoulas. Real-time metric-semantic mapping for autonomous navigation in outdoor environments. *IEEE Transactions on Automation Science and Engineering (T-ASE)*, 2024.

[48] Jianhao Jiao, Jinhao He, Changkun Liu, Sebastian Aegidius, Xiangcheng Hu, Tristan Braud, and Dimitrios Kanoulas. Litevloc: Maplite visual localization for image goal navigation, 2024.