2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Camera Relocalization in Shadow-free Neural Radiance Fields

Shiyao Xu[1,*], Caiyun Liu[1,*], Yuantao Chen[1,2], Zhenxin Zhu[3], Zike Yan[1], Yongliang Shi[1,†], Hao Zhao[1], Guyue Zhou[1]

***Abstract*— Camera relocalization is a crucial problem in computer vision and robotics. Recent advancements in neural radiance fields (NeRFs) have shown promise in synthesizing photo-realistic images. Several works have utilized NeRFs for refining camera poses, but they do not account for lighting changes that can affect scene appearance and shadow regions, causing a degraded pose optimization process. In this paper, we propose a two-staged pipeline that normalizes images with varying lighting and shadow conditions to improve camera relocalization. We implement our scene representation upon a hash-encoded NeRF which significantly boosts up the pose optimization process. To account for the noisy image gradient computing problem in grid-based NeRFs, we further propose a re-devised truncated dynamic low-pass filter (TDLF) and a numerical gradient averaging technique to smoothen the process. Experimental results on several datasets with varying lighting conditions demonstrate that our method achieves state-of-the-art results in camera relocalization under varying lighting conditions. Code and data will be made publicly available.**

I. INTRODUCTION

Camera relocalization is one of the most important problems in computer vision and robotics. Once we construct an accurate scene map out of a dense collection of images captured by drones or vehicles, we aim to recover the camera pose of given images that are taken inside the reconstructed region, facilitating downstream applications [1], [2].

Previous methods utilize discriminative networks [3]–[5] that perform implicit image feature matching and regress the absolute poses on the given images. These methods can recognize the rough places of the test images but fail to achieve pixel-level accuracy.

Recently, neural radiance fields (NeRFs) [6]–[8] have shown their ability to synthesize photo-realistic images. Several works [9]–[15] have used NeRF to refine the inaccurate camera pose inputs from a given initialization (may be produced by absolute pose regression (APR) methods [14]). This line of work optimizes the camera pose by minimizing the photometric error between the rendered image and the observed image. This strategy works well when the lighting conditions remain constant across training and testing sets. Finding the optimal camera pose is then equivalent to rendering the image that can best fit the observation.

However, this equivalence no longer holds in real-world application settings, where there exist lighting changes that cause the scene appearance and shadow regions to change (as shown in Fig. 1-b,c). Under such circumstances, minimizing the rendering error does not guarantee optimal camera poses. A **natural and general** solution may be aligning both sets of images by "normalizing" the scene lighting. In this paper, we find the key factors that break the equivalence that are needed for pose refining NeRFs and propose a "normalizer" that addresses this issue.

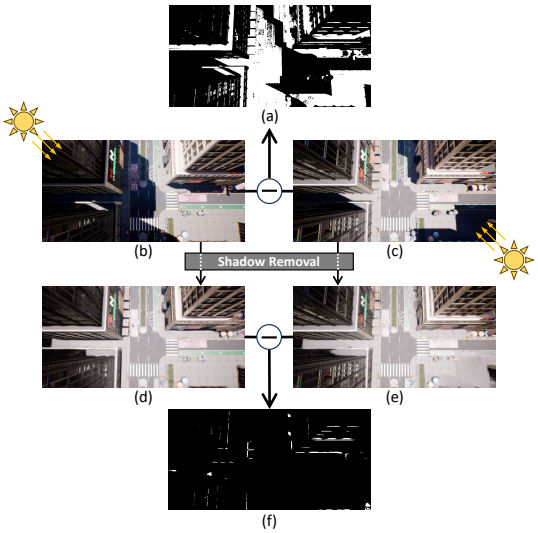

Fig. 1. Illustration on the negative effects caused by shadows in images. (a) demonstrates the image error in the raw images (b and c) taken under different lighting conditions. Directly optimizing pose via rendering error may cause degraded solutions. Our proposed solution first normalizes the images with a shadow removal network. The shadow-free images and the image error are shown respectively in (d)-(f).

As shown in Fig. 1-(a), which illustrates the difference between two images taken at the same camera perspective but with varying lighting conditions, most photometric error occurs in the shadow regions at a given identical camera viewpoint with different lighting conditions. The shadow provides rich image features (e.g., edges, darkness, etc.) when the lighting condition is constant, which helps the convergence but can lead to degraded solutions when lighting changes. In this work, we propose a two-staged pipeline that first aligns both train and test images by a shadow removal module and refines a noisy pose to its global optima robustly in the second stage.

Besides, we implement our neural scene representation with a multi-resolution hash encoding [7]. This representation scales up the expressiveness and reduces the training time of neural radiance fields. However, combining hash-encoded radiance fields with pose optimization is not trivial. Previous works in NeRF-based pose optimization meth-

* Equal contribution. [1] Institute for AI Industry Research, Tsinghua University; [2] Xi'an University of Architecture & Technology; [3] Beihang University.
† Corresponding author. shiyongliang@air.tsinghua.edu.cn
Sponsored by Tsinghua-Toyota Joint Research Fund (20223930097).

ods [10], [14] rely on the (truncated) dynamic low-pass filter (TDLF) that helps optimize the pose in a coarse-to-fine manner by surpassing the high-frequency domains in the NeRF's positional encoding (PE), which is not applicable in hash-encoded NeRFs.

We notice the similarity between the essence of TDLF used on PE and the multiple resolution levels of hash-coded NeRFs [7], where coarse level hash grids correspond to the low-frequency scene structures in positional encoding, while finer level hash grids correspond to the high-frequency scene details in positional encoding. Therefore, we propose to apply a low-pass filter on the weighting to different grid resolution levels. Furthermore, we utilize a numerical gradient averaging technique over the standard autograd operators to encourage smoothened gradient computing.

To sum up, our contributions are:

1) We propose a two-staged pipeline that normalizes the images of various lighting and shadow conditions for camera relocalization.
2) We implement a hash-encoded NeRF for fast training and robust camera pose refinement. A re-devised truncated dynamic low-pass filter and a numerical gradient averaging technique are used to cooperate with the neural scene representation.
3) We propose a new dataset with varying lighting conditions in training and testing sets. We show that our method achieves state-of-the-art results in camera relocalization. Codes and data will be made publicly available.

II. Related Work

A. Camera Relocalization

Classic visual localization methods can be divided into structure-based and image-based. The former uses 2D keypoints in the image to match with 3D points constructed by Structure-from-Motion (SfM) [16] to obtain 2D-3D data associations [17]–[20]. This method gives accurate results but requires the storage of memory-consuming maps and a costly computation. The latter can be realized by image retrieval [21], [22], and is often used for place recognition and loop-closure detection, usually only obtaining a rough position. As deep backbone networks demonstrate powerful feature extraction capabilities, SuperPoint [23] and SiLK [24] learn from self-supervision and outperform classical keypoint detection methods, exhibiting better robustness and generalization ability. These days, CNN-based APR methods [3], [25]–[27] are attracting attention due to their faster speed and better robustness, although the accuracy is not yet sufficiently favorable. PoseNet [3] is the framework of this area, which uses an MLP to regress the camera pose. The later ones improve mainly on the network framework [25], [28]–[30] or training strategy [31], [32]. NeRFs [6] can provide photo-realistic images, and they can also be used for APR tasks. LENS [33] uses a NeRF-W network [34] to synthesize realistic and geometry-consistent images as data augmentation during training and achieves higher localization accuracy. Subsequently, DFNet [4] introduces an online synthetic data generation scheme and proposes a network that extracts domain invariant features in order to reduce the domain gap between synthetic and real images. LATITUDE [14] combines an APR with a pose optimizer to localize in the city but does not take into account the shadow area caused by the scene lighting changes which are very common in outdoor scenes. We address this problem with a shadow-removal pipeline that normalizes all the input images from various lighting conditions.

B. Optimization-based Relocalization

Differentiable rendering methods, such as NeRF, make it possible to recover the camera pose by back-propagating the scene representation. iNeRF [9] proposes the first framework that uses a reconstructed NeRF model to estimate the camera pose. To eliminate the negative impact of positional encoding on pose registration, BARF [10] presents a coarse-to-fine strategy to jointly optimize scene representations and camera poses. Subsequently, [35] uses Gaussian-MLPs to simplify the process of solving the joint task of scene representation and pose optimization. However, the above methods are restricted to small indoor scenes. LATITUDE [14] introduces a two-stage localization mechanism to solve the global localization problem of large-scale scenes. To avoid local optimum, it applies a truncated dynamic low-pass filter during the optimization stage. However, a common problem with the above methods is that they are all MLP-based and still take an extensive amount of time. Many recent works have focused on accelerating the NeRF training, including but not limited to [7], [36], [37]. InstantNGP [7] uses grid sampling and multi-resolution hash encoding to speed up the convergence greatly. In this work, we extend the current pose optimization scheme to be combined with hash-encoded NeRFs.

III. Formulation & Preliminaries

Our goal is to retrieve the accurate camera pose $T(I^{(l')})$ for a given image $I^{(l')}$ with its corresponding lighting condition l' in a pre-reconstructed neural radiance field $\mathcal{F}^{(l)}$ with lighting condition l. This problem is well addressed when $l = l'$ in existing literature such as LATITUDE [14]: first, we pass the test image $I^{(l')}$ through an absolute pose regressor (APR) network, obtaining an initial guess for the pose, denoted as T_0. This initial pose may be noisy due to the implicit nature of CNN-based APR methods. Next, we optimize the inaccurate pose prediction iteratively in a stochastic gradient descent (SGD) manner via:

$$T_{i+1} = T_i - \alpha \cdot \frac{\partial \mathcal{L}(I - \mathcal{F}(T_i))}{\partial T_i}, \tag{1}$$

where α is the learning rate and $\mathcal{L}(I - \mathcal{F}(T_i))$ is the rendering error between the test image and the rendered image by the NeRF network $\mathcal{F}$ at pose T_i.

This optimization pipeline leads to the optimal solution T^* when the lighting condition l remains constant at most times, as described as an equivalence shown in equation 2.

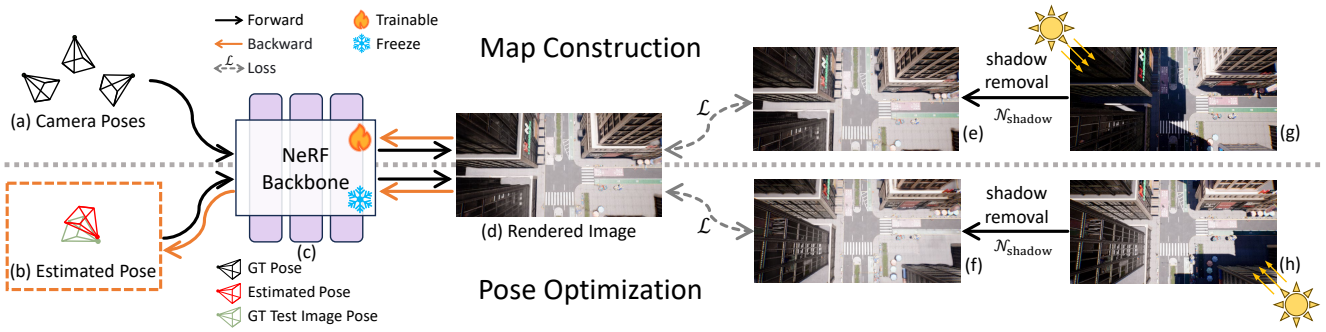

Fig. 2. **Pipeline**. **Top**: In the map construction process, we fit a hash-encoded NeRF onto a set of shadow-free images. **Bottom**: Once the NeRF is trained, we can recover the camera pose for any given test image $I^{(l')}$ by first processing the image with the same shadow removal network $\mathcal{N}_{\text{shadow}}$ as used in the training stage and refine the initial pose recursively with the NeRF network fixed.

However, when $l \neq l'$, the optimization scheme in equation 1 cannot lead to an optimal solution since $\mathcal{L}(I^{(l')} - \mathcal{F}^{(l)}(T^*))$ may not be the minimal error anymore.

$$\mathcal{L}(I - \mathcal{F}(T^*)) = 0. \tag{2}$$

This photometric inconsistency causes the sensitive feature matching steps of equation 1 to fail. This key observation then leads us to a natural solution of aligning all images to a "normalized" lighting condition l_0 by a normalizer $\mathcal{N}(I^{(l_*)}) = I^{(l_0)}$. DFNet [4] proposes to align the images with a histogram-assisted NeRF. However, as shown in Fig. 1, the photometric error caused by lighting changes is primarily due to shadow changes.

Our two-staged pipeline addresses this issue by aligning the training set images $\{I_k^{(l)}\}$ and the test image $I^{(l')}$ with a shadow removal network $\mathcal{N}_{\text{shadow}}$. In the first stage, we train a neural radiance field $\mathcal{F}^{(l_0)}$ with normalized images. Then, in the second stage, we optimize the test camera pose by minimizing $\mathcal{L}(\mathcal{N}_{\text{shadow}}(I^{(l')}) - \mathcal{F}^{(l_0)}(T_i))$.

IV. Method

A. Pipeline

As illustrated in Fig. 2, our proposed method is structured as a two-stage pipeline: 1. We reconstruct the scene map $\mathcal{F}$ (top row in Fig. 2) with normalized lighting conditions l_0 by pre-processing the training images with a shadow removal network $\mathcal{N}_{\text{shadow}}$; 2. We find the accurate camera pose T of the test image $I^{(l')}$ (bottom row in Fig. 2). For the scene map, we construct a three-dimensional neural scene map based on a multi-resolution hash grid using a set of posed RGB images.

It's noteworthy that the images used during the map construction and pose optimization stages contain various shadows. Since the shadow variations as one major manifestation of photometric inconsistency (as shown in Fig. 1-a), we employ the shadow removal network, denoted as $\mathcal{N}_{\text{shadow}}$, for images used in NeRF map construction and pose optimization, to obtain "normalized" shadow-free image components, further satisfying the equivalence in equation 2.

Shadow Removal: In our image shadow removal process, there are two distinct steps: the first is shadow detection, and the second is shadow removal reliant on the shadow mask produced in the first step. For shadow detection procedure $\mathcal{D}_{\text{shadow}}$, we utilized MTMT [38], which segments shadow in the input images $I^{(l_*)}$, yielding a shadow region mask M. For the shadow removal process $\mathcal{R}_{\text{shadow}}$, we employed a Transformer-based network following [39]. By inputting the previously obtained mask and the input image, it associates the shadowed regions with non-shadowed regions to remove shadows within the masked area, producing a shadow-free image. The shadow removal process can be expressed as:

$$I^{(l_0)} = \mathcal{N}_{\text{shadow}}(I^{(l_*)}) = \mathcal{R}_{\text{shadow}}(I^{(l_*)}, \mathcal{D}_{\text{shadow}}(I^{(l_*)})). \tag{3}$$

Map Construction: In this section, we detailed our hash-encoded scene representation $\mathcal{F}$. The scene representation network is approximated with a multi-resolution hash grid, along with a shallow MLP decoder. Given a 3D position $\mathbf{x} \in \mathbb{R}^3$. We query the hash grid in each of the L resolution levels to obtain the hash feature $\{he_k(\mathbf{x})\}_{k=1}^{L}$.

The hash encoding $he_k(\mathbf{x})$ at the k^{th} resolution level is derived from tri-linear interpolation on the 8 neighboring grid points around the queried position:

$$he_k(\mathbf{x}) = \texttt{interp}_k\left(\left(\bigoplus_{i=1}^{3} x_i\pi_i\right) \bmod T\right), \tag{4}$$

where $\texttt{interp}_k(\cdot)$ denotes the tri-linear interpolation operator in the k^{th} resolution level grid, π_i and T are the parameters of the hash function.

For each resolution grid, we obtain an F-dimensional feature vector. Subsequently, the obtained L feature vectors are sequentially concatenated:

$$HE(\mathbf{x}) = (he_1(\mathbf{x}), he_2(\mathbf{x}), \cdots, he_L(\mathbf{x})). \tag{5}$$

We employ a shallow MLP to predict the sampled point color and density from the extracted hash feature $HE(\mathbf{x})$ and direction $\mathbf{d} \in \mathbb{R}^3$ to the volume density σ of the sample point and its color $\mathbf{c}$. We follow the approach of Instant-NGP [7] and utilize spherical harmonics $SH(\cdot)$ for direction encoding. The forwarding pass of our NeRF network can be described as:

$$\mathbf{c}, \sigma = \mathcal{F}(\mathbf{x}, \mathbf{d}) = \text{MLP}(HE(\mathbf{x}), SH(\mathbf{d})). \tag{6}$$

Subsequently, the rendered image $\hat{I}^{(l_0)}$ is obtained using volume rendering. To achieve the alignment of training images from their original lighting condition to the shadow-free lighting condition l_0, the loss function we employ during the NeRF training phase is:

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left| \hat{I}^{(l_0)}(\mathbf{r}) - \mathcal{N}_{\text{shadow}}(I^{(l_*)}(\mathbf{r})) \right|_1, \tag{7}$$

where $\mathcal{R}$ represents a batch of rays obtained through sampling during the training process, $\mathbf{r}$ is a ray from the set $\mathcal{R}$, and $I^{(l_*)}$ denotes the images from the training set with an arbitrary lighting condition l_*.

This loss function compels the network's rendering output to closely resemble the shadow-free training images. Consequently, the constructed map represents a normalized scene devoid of shadows, laying a solid foundation for pose optimization. During the pose optimization stage in section IV-B, the same shadow removal model is utilized to remove shadows from the test images, ensuring that both the map and the images used for pose optimization are aligned to l_0.

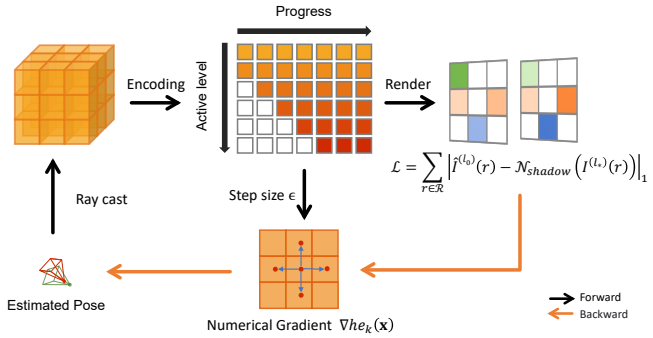

Fig. 3. Illustration on the **re-devised TDLF** and the **numerical gradient averaging** techniques. During the pose optimization stage, we filter the extracted feature on its frequency domain by using a set of closed-form weighting parameters. When updating the estimated pose with SGD, we use numerical gradient averaging over the standard Autograd operators to ensure a smooth optimization process.

B. Pose Optimization

Pose optimization in NeRFs involves refining the camera's pose to align the virtual and real-world scenes accurately. Given a shadow-free image $I^{(l_0)}$ and corresponding initial pose guess $\hat{T}_0$, the goal is to minimize the photometric loss between the observed image and the rendered image $\hat{I}$ which is obtained from camera pose. We optimize the camera pose on the manifold and express the parameter of the camera extrinsic as $\xi \in \mathfrak{se}(3)$, the optimized pose is obtained by equation 8 as:

$$\hat{T} = \exp(\hat{\xi})\hat{T}_0. \tag{8}$$

To optimize the pose parameters, we use the same loss function as in the map reconstruction stage (equation 7).

However, trivially minimizing the loss function may lead to sub-optimal results. We further propose a coarse-to-fine optimization strategy that is in the same spirit as the dynamic low-pass filters proposed in [10], [14] and a numerical gradient averaging technique which smoothens the optimization process, as shown in Fig. 3.

Coarse-to-fine Optimization: Since the input images are rich in high-frequency signals, using gradient descent to minimize photometric loss can lead to local minima rather than finding the global minimum. A coarse-to-fine smoothing technique (referred to as the truncated dynamic low-pass filter, or the TDLF) had previously been proposed [10], [14], in which the feature of position encoding is used to separate the high- and low-frequency image components of the scene and the high-frequency details are smoothed out in the early stage of pose estimation. The TDLF are implemented by suppressing the high-frequency components of the positional encoding (PE). Since PE is no longer applicable in our grid-based scene representation, it is not trivial to utilize TDLF in our setting. In this paper, we extend the TDLF to grid-based NeRFs by appending weighting parameters on the extracted feature from each resolution level as:

$$HE(\mathbf{x}) = (\omega_1(\alpha)he_1(\mathbf{x}), \cdots, \omega_L(\alpha)he_L(\mathbf{x})), \tag{9}$$

where $\alpha \in [0, L]$ is a function of the optimization progress. While a pre-trained NeRFs model is given, rendering with too few resolution levels activated can lead to invalid outputs, we set an appropriate initial α to avoid this problem by setting $\alpha = \min(\alpha_0 + \text{progress}, 1)L$. The weighting parameter for level k is set to:

$$w_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ \frac{1-\cos((\alpha-k)\pi)}{2} & \text{if } 0 \le \alpha - k < 1 \\ 1 & \text{if } \alpha - k \ge 1 \end{cases}. \tag{10}$$

Numerical gradient averaging: The numerical gradient averaging technique aims to solve the derivative discontinuity problem caused by hash encoding. From the loss function described in equation 7 described previously, we can derive the steepest descent direction for pose variables:

$$\mathbf{J}(\mathbf{u};\xi) = \sum_{i=1}^{N} \frac{\partial g(\mathbf{y}_1, \ldots, \mathbf{y}_N)}{\partial \mathbf{y}_i} \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_i(T)} \frac{\partial \mathcal{W}(T)}{\partial \xi}, \tag{11}$$

where g represents the volume rendering process, $\mathbf{x}_i$ is the 3D point along the sampling ray with corresponding hash encoding $\mathbf{y}_i$ and $\mathcal{W}$ represents the ray casting process.

Hash encoding has localized derivatives, meaning that when points cross grid cell boundaries, the resulting hash entries will change discontinuously, resulting in disrupt changes in $\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_i(T)}$. This further blocks the feature-matching process that is vital for the pose optimization process. Inspired by recent work in neural 3D reconstruction [40], we introduce a numerical gradient averaging technique to allow for feature sharing among neighboring grid cells.

To compute the numerical gradient, we sample a set of additional points around the queried points. To be specific, given a 3D point $\mathbf{x}_i = (x_i, y_i, z_i)$ on a sampling ray, we need to sample two points along each axis of canonical coordinates around $\mathbf{x}_i$ with a step size of ϵ. For example,

the x-component of $\nabla he_k(\mathbf{x}_i)$ can be found as

$$\nabla_x he_k(\mathbf{x}_i) = \frac{he_k\left(\mathbf{x}_i + \boldsymbol{\epsilon}_x\right) - he_k\left(\mathbf{x}_i - \boldsymbol{\epsilon}_x\right)}{2\epsilon}, \qquad (12)$$

where $\epsilon_x = [\epsilon, 0, 0]$, ϵ is the inverse of the resolution of the currently activated level L_{act} which is controlled by α with $L_{\text{act}} = \lceil \alpha \rceil$. In total, 6 additional points are sampled to compute the full numerical gradient.

Although the numerical gradient would be equivalent to the analytical gradient if the step size is smaller than the grid size, the numerical gradient can still smooth gradients when $\mathbf{x}_i$ near the borders as $\mathbf{x}_i \pm \boldsymbol{\epsilon}$ can move across the grids.

V. Experiments and Analysis

A. Dataset

We evaluate our method using the New York scene in our proposed Shadow Urban Minimum Altitude Dataset (SUMAD) and partial scenes from the public dataset NeRF-OSR [41]. The SUMAD is a virtual-scene dataset made by the simulator AirSim [42] built on top of the Unreal Engine. To replicate various real-world shadowing scenarios, we manipulate the direction and position of the light source within Unreal Engine, creating three distinct types of lighting conditions for each scene. Further, to enhance data realism, we incorporate two authentic city scene models within Unreal Engine, faithfully replicating urban environments resembling New York and San Francisco. It is worth noting that we are the first to release a dataset featuring diverse shadowing scenarios within large-scale city environments. The NeRF-OSR data is a set of real-world data captures that contain different lighting conditions.

B. Implementation Details

We used a hash grid configuration following Instant-NGP [7]. We train the scene for 100,000 iterations. The base MLP consists of 1 hidden layer of 64 units and output 16 channels which together with the direction encoded by the harmonic function forms the input to the head MLP, which consists of 2 hidden layers of 64 units. We resize the images to 960 × 480 pixels and randomly cast rays during the training steps, with constrain on the amount of 3D sampling points to be limited to $1 << 18$. We use an Adam optimizer with an initial learning rate of 1×10^{-2} decaying exponentially to 1×10^{-4} for scene reconstruction and 1.2×10^{-2} to 1.2×10^{-3} for pose optimize, while a scaler is applied to magnify loss by 2^{10}.

In order to implement the coarse-to-fine strategy in the hash grid encoding and numerical gradient, we add a deterministic layer after hash grid encoding, which sets the active level and step size of the numerical gradient. The active level is initially set to 8 in the experiments. All experiments are conducted on a Linux system with an NVIDIA RTX3090 GPU with 24GB of memory, and 1000 iterations are run for each optimization.

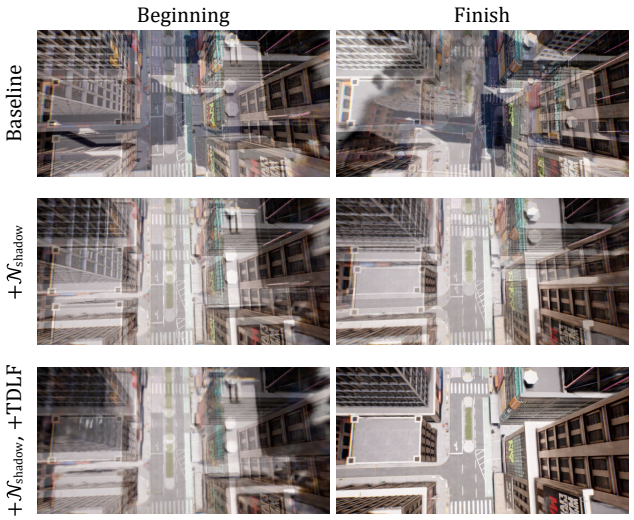

Fig. 4. We show the optimization process of our methods with different design choices. The rendered and observed images are stacked to demonstrate both rendering quality and pose accuracy.

C. Results

The quantitative evaluations on the SUMAD dataset are shown in Table I. During experiments, we introduce a 4-meter translation error parallel to the trajectory forward direction to the ground truth pose as an initial pose for all methods and evaluate all of the methods on image synthesis results and the translation/rotation errors on the recovered poses. Note that the series of experiments with DFNet are not applicable for introducing initial pose errors and for evaluating image synthesis qualities. The image synthesis metrics (PSNR, SSIM, and LPIPS [43]) are evaluated on the shadow-free images no matter whether the shadow removal process is used or not for fair comparisons.

The baseline methods evaluated here are: 1. DFNet [4] trained on the raw images and evaluated on test images with different lighting conditions, which causes a degraded solution in regressing the absolute position of the test images; 2. DFNet variant that is trained and evaluated on the shadow-free images shows a significant improvement in localization accuracy (from 491m to 6m); 3/4. LATITUDE [14] trained on raw/shadow-free images; 5. Performing iNeRF-like [9] direct pose refinement on an InstantNGP [7]; 6/7. Our proposed method with and without shadow removal process.

As can be inferred from the comparisons between shadow-free and unprocessed counterparts, the shadow removal process significantly improves the image synthesis quality and the relocalization accuracy, no matter the scene representation used (DFNet [4], LATITUDE [14], and our proposed hash-encode NeRF). Among all the evaluated baseline methods, our proposed two-staged pipeline achieves the best image synthesis quality, the most accurate poses recovered, and the fastest overall time used for training and evaluation.

D. Ablation Study

Our method uses a coarse-to-fine strategy and numerical gradient averaging technique to achieve a more accurate and

TABLE I

QUANTITATIVE RESULTS WITH BASELINE METHODS.

Methods	Translation Error(m)	Rotation Error(°)	PSNR↑	SSIM↑	LPIPS↓	Time Train / Inference
DFNet [4]	491.457	30.315	-	-	-	40h / $<$ 1s
DFNet ($+\mathcal{N}_{\text{shadow}}$)	6.002	28.654	-	-	-	40h / $<$ 1s
LATITUDE [14]	12.174	11.803	9.260	0.207	0.587	10h / 5.5m
LATITUDE ($+\mathcal{N}_{\text{shadow}}$)	0.094	0.534	24.028	0.758	**0.195**	10h / 5.5m
iNeRF+iNGP [7], [9]	8.697	8.832	9.766	0.248	0.635	1.5h / 27s
Ours ($-\mathcal{N}_{\text{shadow}}$)	6.747	8.398	9.831	0.219	0.636	1.5h / 55s
Ours	**0.091**	**0.106**	**25.799**	**0.826**	0.253	1.5h / 55s

TABLE II

ABLATION STUDY WITH DIFFERENT INITIAL TRANSLATION ERRORS.

Initial Error(m)	Numerical Gradient Averaging	TDLF	Translation Error(m)	Rotation Error(°)
4	×	×	0.12	0.13
	×	✓	0.12	0.13
	✓	×	0.09	0.11
	✓	✓	**0.09**	**0.11**
8	×	×	3.42	1.22
	×	✓	1.18	0.27
	✓	×	0.09	0.10
	✓	✓	**0.09**	**0.10**
12	×	×	4.20	1.44
	×	✓	4.20	1.20
	✓	×	1.85	1.69
	✓	✓	**0.10**	**0.11**
16	×	×	11.05	4.68
	×	✓	10.09	4.37
	✓	×	3.71	4.13
	✓	✓	**0.07**	**0.09**

TABLE III

ABLATION STUDY WITH DIFFERENT INITIAL ROTATION ERROR.

Initial Error(°)	Numerical Gradient Averaging	TDLF	Translation Error(m)	Rotation Error(°)
4	×	×	0.11	0.13
	×	✓	0.10	0.11
	✓	×	0.10	0.10
	✓	✓	**0.10**	**0.10**
8	×	×	0.32	0.22
	×	✓	0.10	0.11
	✓	×	2.58	3.37
	✓	✓	**0.10**	**0.10**
12	×	×	2.34	1.86
	×	✓	3.18	3.55
	✓	×	3.18	2.79
	✓	✓	**1.06**	**0.45**
16	×	×	7.75	5.41
	×	✓	9.07	5.92
	✓	×	5.22	3.69
	✓	✓	**4.34**	**3.60**

robust pose optimization process. To evaluate the effects of our proposed improvements, we perform an ablation study that quantitatively analyzes how the error tolerance and convergence accuracy of pose optimization change when the improvements are applied or not.

Experiments are carried out at six randomly selected positions in the SUMAD dataset, with initial translation or rotation errors of the same size introduced at each position. The average error results are recorded in table II and table III. The results presented in the tables demonstrate that our proposed method exhibits the greatest robustness to both translation and rotation perturbations, and also achieves the highest convergence accuracy out of all methods tested. The results validate the effectiveness of incorporating the coarse-to-fine strategy into state estimation methods based on Grid-based NeRFs. The coarse-to-fine strategy can help the pose optimization to escape from local minima generated by the high-frequency information, as shown in Fig. 4. Besides, numerical gradient also shows its superiority compared to analytical gradient in the context of hash grid encoding even without the coarse-to-fine strategy, which is consistent with what is expected in section IV-B. We also conducted tests using our method on partial scenes from NeRF-OSR and obtained relatively good results, as shown in Fig. 5.

(a) (b)

Fig. 5. Result of our method optimizing pose on the NeRF-OSR dataset. (a) Original image from dataset at the ground truth pose. (b) Images rendered with optimized noisy pose.

VI. CONCLUSIONS

In this paper, we address the challenge of camera pose refinement under varying lighting conditions. We've pinpointed shadow differences as a principal source of photometric errors, highlighting the necessity of consistent lighting for accurate pose refinement in NeRFs. Our proposed two-staged pipeline ensures images are normalized irrespective of shadow and lighting variations. By introducing a shadow removal module, we've successfully bridged the photometric discrepancies between images. Further, the integration of multi-resolution hash encoding with our neural scene representation has not only amplified its expressiveness but has also substantially expedited the training process. We've ingeniously combined this with pose optimization, offering a method that is both accurate and efficient. Our method achieves state-of-the-art results on our proposed dataset and other public dataset.

References

[1] E. Šlapak, E. Pardo, M. Dopiriak, T. Maksymyuk, and J. Gazda, *Neural radiance fields in the industrial and robotics domain: Applications, research opportunities and use cases*, 2023. (visited on 08/15/2023).

[2] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-Only Robot Navigation in a Neural Radiance World," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[3] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[4] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, *DFNet: Enhance Absolute Pose Regression with Direct Feature Matching*, 2022. (visited on 09/08/2023).

[5] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4762–4769.

[6] 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," in *Computer Vision – ECCV 2020*, 2020, pp. 405–421.

[7] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, 2022. (visited on 10/05/2022).

[8] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A Modular Framework for Neural Radiance Field Development," *ACM Transactions on Graphics*, vol. 1, no. 1, 2023. (visited on 06/24/2023).

[9] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting Neural Radiance Fields for Pose Estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1323–1330.

[10] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-Adjusting Neural Radiance Fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751. (visited on 10/13/2022).

[11] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "NoPe-NeRF: Optimising Neural Radiance Field With No Pose Prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169. (visited on 06/11/2023).

[12] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-Calibrating Neural Radiance Fields," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5826–5834.

[13] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, *NeRF–: Neural Radiance Fields Without Known Camera Parameters*, 2022. (visited on 05/06/2023).

[14] Z. Zhu, Y. Chen, Z. Wu, C. Hou, Y. Shi, C. Li, P. Li, H. Zhao, and G. Zhou, "LATITUDE: Robotic Global Localization with Truncated Dynamic Low-pass Filter in City-scale NeRF," in *2023 IEEE Conference on Robotics and Automation (ICRA 2023)*, 2022. (visited on 10/05/2022).

[15] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, *Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields*, 2022. (visited on 03/26/2023).

[16] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[17] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1455–1461, 2016.

[18] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[19] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic match consistency for long-term visual localization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 383–399.

[20] L. Liu, H. Li, and Y. Dai, "Efficient global 2d-3d matching for camera localization in a large-scale 3d map," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2372–2381.

[21] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, Springer, 2016, pp. 241–257.

[22] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[24] P. Gleize, W. Wang, and M. Feiszli, "Silk–simple learned keypoints," *arXiv preprint arXiv:2304.06194*, 2023.

[25] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 627–637.

[26] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2616–2625.

[27] F. Xue, X. Wu, S. Cai, and J. Wang, "Learning multi-view camera relocalization with graph neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 11 372–11 381.

[28] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5644–5651.

[29] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 870–877.

[30] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2713–2722.

[31] S. Chen, Z. Wang, and V. Prisacariu, "Direct-posenet: Absolute pose regression with photometric consistency," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 1175–1185.

[32] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6555–6564.

[33] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Lens: Localization enhanced by nerf synthesis," in *Conference on Robot Learning*, PMLR, 2022, pp. 1347–1356.

[34] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7206–7215.

[35] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey, "Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation," in *European Conference on Computer Vision*, Springer, 2022, pp. 264–280.

[36] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance Fields without Neural Networks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5491–5500.

[37] C. Sun, M. Sun, and H.-T. Chen, "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5449–5459.

[38] Z. Chen, L. Zhu, L. Wan, S. Wang, W. Feng, and P.-A. Heng, "A multi-task mean teacher for semi-supervised shadow detection," in *CVPR*, 2020.

[39] L. Guo, S. Huang, D. Liu, H. Cheng, and B. Wen, "Shadowformer: Global context helps image shadow removal," *arXiv preprint arXiv:2302.01650*, 2023.

[40] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-Fidelity Neural Surface Reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8456–8465. (visited on 06/04/2023).

[41] V. Rudnev, M. Elgharib, W. Smith, L. Liu, V. Golyanik, and C. Theobalt, "NeRF for Outdoor Scene Relighting," 2022. (visited on 10/25/2022).

[42] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: `https://arxiv.org/abs/1705.05065`.

[43] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.