2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Parallel Inversion of Neural Radiance Fields for Robust Pose Estimation

Yunzhi Lin[1,2], Thomas Müller[1], Jonathan Tremblay[1], Bowen Wen[1], Stephen Tyree[1],
Alex Evans[1], Patricio A. Vela[2], Stan Birchfield[1]
[1]NVIDIA: {tmueller, jtremblay, bowenw, styree, alexe, sbirchfield}@nvidia.com
[2]Georgia Institute of Technology: {yunzhi.lin, pvela}@gatech.edu

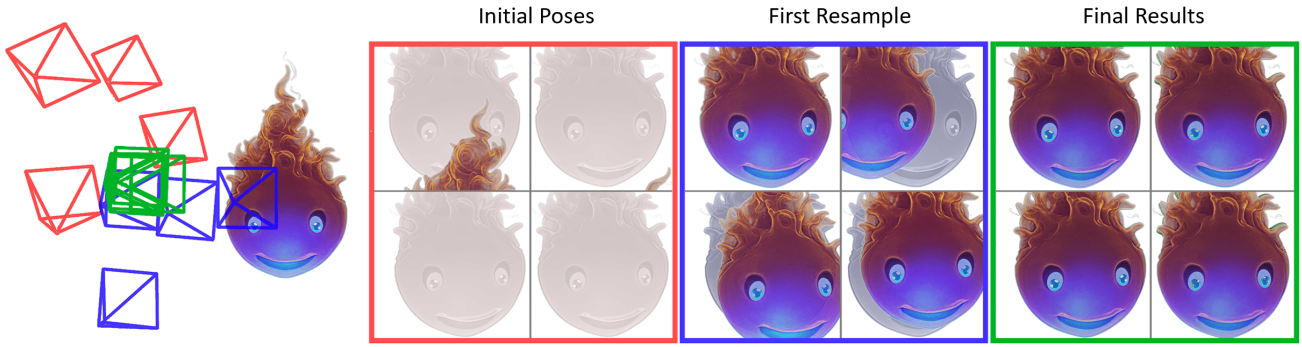

Fig. 1. *Our NeRF-based parallelized optimization method estimates the camera pose from a monocular RGB image of a novel object.* The optimization iteratively updates a set of pose estimates in parallel by backpropagating discrepancies between the observed and rendered image. LEFT: To simplify the display, we show four camera hypotheses at three iterations: initial (red), after the first resample (blue), and final (green). RIGHT: Corresponding renderings of estimated poses (in color) overlaid on the observed image (grayscale).

***Abstract*— We present a parallelized optimization method based on fast Neural Radiance Fields (NeRF) for estimating 6-DoF pose of a camera with respect to an object or scene. Given a single observed RGB image of the target, we can predict the translation and rotation of the camera by minimizing the residual between pixels rendered from a fast NeRF model and pixels in the observed image. We integrate a momentum-based camera extrinsic optimization procedure into Instant Neural Graphics Primitives, a recent exceptionally fast NeRF implementation. By introducing parallel Monte Carlo sampling into the pose estimation task, our method overcomes local minima and improves efficiency in a more extensive search space. We also show the importance of adopting a more robust pixel-based loss function to reduce error. Experiments demonstrate that our method can achieve improved generalization and robustness on both synthetic and real-world benchmarks.**

I. INTRODUCTION

6-DoF pose estimation—predicting the 3D position and orientation of a camera with respect to an object or scene—is a fundamental step for many tasks, including some in robot manipulation and augmented reality. While RGB-D or point cloud-based methods [1]–[4] have received much attention, monocular RGB-only approaches [5], [6] have great potential for wider applicability and for handling certain material properties that are difficult for depth sensors—such as transparent or dark surfaces.

Much research in this area has focused on instance-level object pose estimation [7]–[11]. These methods assume that a textured 3D model of the target object is available for training. Such methods have achieved success under different scenarios but suffer from a lack of scalability. Research beyond this limitation considers category-level object pose estimation [6], [12]–[14]. These methods scale better for real-world applications, since a single trained model works for a variety of object instances within a known category. Nevertheless, the effort needed to define and train a model for each category remains a limitation. For more widespread generalizability, it is important to be able to easily estimate poses for arbitrary objects.

The emergence of Neural Radiance Fields (NeRF) [15] has the potential to facilitate novel object pose estimation. NeRF and its variants learn generative models of objects from pose-annotated image collections, capturing complex 3D structure and high-fidelity surface details. Recently, iNeRF [16] has been proposed as an analysis-by-synthesis approach for pose estimation built on the concept of inverting a NeRF model. Inspired by iNeRF's success, this paper further explores the idea of pose estimation via neural radiance field inversion.

A drawback of NeRF is its computational overhead which impacts execution time. To overcome this limitation, we leverage our fast version of NeRF, known as Instant Neural Graphics Primitives (Instant NGP) [17]. Using Instant NGP model inversion provides significant speedups over NeRF. The structure of Instant NGP admits parallel optimization, which is leveraged to overcome issues with local minima and thereby achieve greater robustness than possible with iNeRF. Similar to iNeRF, our pose estimation requires three inputs: a single RGB image with the target, an initial rough pose estimate of the target, and an instant NGP model trained from multiple views of the target.

Considering that a single camera pose is vulnerable to local minima during optimization iterations, we leverage parallelized Monte Carlo sampling. At adaptive intervals, camera pose hypotheses are re-sampled around the hypothe-

Project page: https://pnerfp.github.io

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161117

ses with the lowest loss. This design alleviates the issue of convergence to local minima and improves efficiency of search over a more extensive search space.

The gradients of pixel residuals calculated between the rendered model and the target view are backpropagated to generate camera pose updates. Unlike iNeRF where a subsample of a new image is rendered at each iteration,we enable hundreds of thousands of rays to work independently in parallel to accumulate gradient descent updates per camera pose hypothesis. This design dramatically improves the efficiency. Furthermore, we investigate different pixel-based loss functions to identify which approach to quantifying the visual difference between the rendered model and the observed target image best informs the camera pose updates. As shown in the ablation study, the mean absolute percentage error (MAPE) [18] loss exhibits better robustness to disturbances.

In summary, this work makes the following contributions:

- A parallelized, momentum-based optimization method using NeRF models is proposed to estimate 6-DoF poses from monocular RGB input. The object-specific NeRF model does not require pre-training on large datasets.
- Parallelized Monte Carlo sampling is introduced into the pose estimation task, and we show the importance of pixel-based loss function selection for robustness.
- Quantitative demonstration through synthetic and real-world benchmarks that the proposed method has improved generalization and robustness.

II. Related Works

Neural 3D Scene Representation. Recent works [19]–[21] have investigated representing 3D scenes implicitly with neural networks, where coordinates are sampled and fed into a neural network to produce physical field values across space and time [22]. NeRF [15] is a milestone approach demonstrating that neural scene representations have the capabilities to synthesize photo-realistic views. Since then, significant effort has been put into pushing the boundaries of NeRF. Follow-up works have focused on speeding up the training and inference processes [17], [23], [24], adding support for relighting [25], relaxing the requirement of known camera poses [26], [27], reducing the number of training images [28], extending to dynamic scenes [29], and so on. NeRF also opens up opportunities in the robotics community. Researchers have proposed to use it to represent scenes for visuomotor control [30], reconstruct transparent objects [31], generate training data for pose estimators [32] or dense object descriptors [33], and model 3D object categories [34]. In this work, we aim to follow in their footsteps by applying NeRF directly to the 6-DoF pose estimation task.

Generalizable 6-DoF Pose Estimation. Generalizable 6-DoF pose estimation—not limited to any specific target or category—from RGB images has been a long-standing problem in the community. Existing methods tend to share a similar pipeline of two phases: 1) model registration and 2) pose estimation.

Traditional methods [35]–[38] first build a 3D CAD model via commercial scanners or dense 3D reconstruction techniques [39], [40]. They resolve the pose by finding 2D-3D correspondences (via hand-designed features like SIFT [41] or ORB [42]) between the input RGB image and the registered model. However, creating high quality 3D models is not easy, and finding correspondence across a large database (renderings from different viewpoints) can be time-consuming [35]. More recently, several attempts have been made to revisit the object-agnostic pose estimation problem with deep learning. The presumption is that a deep network pretrained on a large dataset can generalize to find correspondence between the query image and the registered model for novel objects. OnePose [43], inspired by visual localization research, proposes to use a graph attention network to aggregate 2D features from different views during the registration phase of structure-from-motion [44]. Then the aggregated 3D descriptor is matched with 2D features from the query view to solve the PnP problem [45]. Similarly, OSOP [46] explores solving the PnP problem with a dense correspondence between the query image and the coordinate map from a pre-built 3D CAD model. On the other hand, Gen6D [47] only needs to register the model with a set of posed images. Following the iterative template matching idea [48], [49], its network takes as input several neighboring registered images closest to the predicted pose and repeatedly refines the result.

While data-driven approaches rely on the generalization of a large training dataset (usually composed of both synthetic & real-world data) [47], iNeRF [16] is an optimization on-the-fly approach free of pretraining. Each new object is first registered by a NeRF model [15], after which iNeRF can optimize the camera pose on the synthesized photo-realistic renderings from NeRF. Although iNeRF's idea seems promising, there still remain several challenges. The first is the expensive training cost of a NeRF model, which may take hours for just one target. Additionally, iNeRF's pose update strategy is inefficient, as the accumulation and backpropagation of the loss gradient is performed until a subsample of a new image is rendered. Moreover, the optimization process of a single pose hypothesis is easily trapped in local minima due to outliers. To deal with the aforementioned issues, we propose a more efficient and robust approach leveraging the recent success of Instant NGP [17]. We re-formulate the camera pose representation as the Cartesian product $\mathrm{SO}(3) \times \mathrm{T}(3)$ and integrate the optimization process into the structure of Instant NGP. We also adopt parallelized Monte Carlo sampling to improve robustness to local minima. Loc-NeRF [50] is another concurrent work using Monte Carlo sampling to improve iNeRF.

III. Preliminaries

NeRF. Given a collection of N RGB images $\{I_i\}_{i=1}^{N}, I_i \in [0,1]^{H\times W\times 3}$ with known camera poses $\{T_i\}_{i=1}^{N}$, NeRF [15] learns to represent a scene as 5D neural radiance fields (spatial location and viewing direction). It can synthesize novel views by querying 5D coordinates along the camera rays and use classic volume rendering techniques to project the output colors and densities into an image.

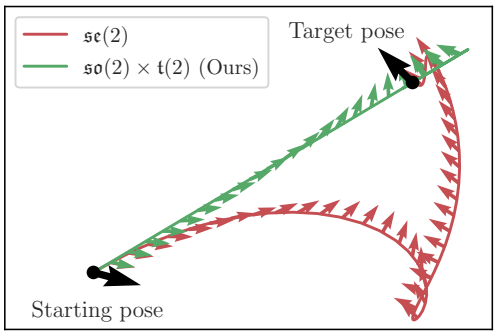

Fig. 2. GREEN: Our decomposition of gradients and their momentum into the rotational Lie algebra $\mathfrak{so}(2)$ and the translational Lie algebra $\mathfrak{t}(2)$ yields a straight path from the starting pose to the target pose using gradient descent with momentum, converging in 69 steps. Momentum causes our optimization to overshoot and snap back to the target, all along the straight line. RED: Using the special Euclidean Lie algebra $\mathfrak{se}(2)$ [16], [26] leads to a suboptimal curved path due to coupling between translation and rotation, requiring 85 steps to converge.

Instant NGP. To further reduce the training and inference cost of the vanilla NeRF [15], Instant NGP [17] proposes to adopt a small neural network augmented by a multi-resolution hash table of trainable feature vectors. This structure allows the network to disambiguate hash collisions, making it easy to parallelize on GPUs. The method achieves a combined speedup of several orders of magnitude, allowing its use in time-constrained settings like online training and inference.

iNeRF's Formulation. Assuming the target scene has been trained with a NeRF model parameterized with weight Θ and the camera intrinsics are known, iNeRF [16] aims to recover the camera pose $T \in SE(3)$ of an observed image I given the weight Θ:

$$\hat{T} = \underset{T \in \mathrm{SE}(3)}{\operatorname{argmin}} \mathcal{L}(T \mid I, \Theta) \tag{1}$$

with $\mathcal{L}$ the loss between the NeRF rendering and the observed image. It uses L2 loss in practice. In the optimization process, iNeRF fixes the NeRF's weight Θ and iteratively updates T to minimize $\mathcal{L}$.

IV. APPROACH

A. Momentum-based Camera Extrinsics Optimization

For this work, we modified the Instant NGP [17] camera pose and gradient representations from their standard use in NeRF. Critically, this permitted the dynamics of the gradient updates to incorporate momentum-based approaches for enhanced optimization. The section details those changes.

a) Camera Pose Representation: Camera poses consist of a translation component (position) as well as a rotation component (orientation) and are often modeled by the special Euclidean group in 3D, $SE(3)$. The goal of extrinsics optimization in NeRF [16], [26] is to find those camera poses that minimize the image-space loss by gradient descent. Gradient updates are computed in the special Euclidean Lie algebra $\mathfrak{se}(3)$, then applied to generate a camera pose update combining rotation and translation. However, using a native $SE(3)/\mathfrak{se}3$ representation has a disadvantage: a camera pose update's center of rotation is not at the camera origin, but on the screw axis, which couples camera position and orientation. This coupling can lead to nonintuitive optimization trajectories when momentum is involved. To decouple the translation and rotation updates, we model camera pose as the Cartesian product $\mathrm{SO}(3) \times \mathrm{T}(3)$ (and likewise the respective Lie algebra, $\mathfrak{so}(3) \times \mathfrak{t}(3)$), which employs an additive structure on $\mathrm{T}(3)$ and a product structure on $\mathrm{SO}(3)$. Momentum in this representation moves in straight lines and rotates along geodesic paths over the surface of the sphere, converging in fewer steps.

Fig. 2 shows a 2D example, where we use the momentum-based Adam optimizer [51] to minimize the mean squared error (MSE) between the 3×3 matrices describing the target pose and the current pose. The gradient of the MSE always points along the straight line to the target in 2D space, yet the trajectory through $\mathfrak{se}(2)$ (shown in red) deviates from the straight line, because of the aforementioned coupling. Note that the trajectory *starts out* straight—gradients without momentum always point straight to the target, regardless of representation—begins to curve once momentum builds up, which becomes increasingly inaccurate as $SE(2)$ is traversed. The trajectory eventually converges as the momentum is corrected over the course of many optimization steps.

b) Momentum-Based Optimization: Contemporary momentum-based optimization has empirically demonstrated more effective convergence properties over standard gradient-based approaches, especially when combined with an adaptive update law. For implementation of the optimization update laws, the Adam optimizer [51] with first and second moments is applied independently on the two subspaces of our representation, $\mathrm{SO}(3)$ and $\mathrm{T}(3)$. Crucially, the momentum-based updates are cheaply and stably computed from the NeRF implementation. In NeRF, each pixel corresponds to a ray with origin o and direction d, along which the model is evaluated at K positions, $\mathrm{p}_i = \mathrm{o} + t_i \cdot \mathrm{d}$, based on moving distances t_i along the ray. In our decomposition, the ray origin o is the same as the camera origin, and thus corresponds to $\mathrm{T}(3)$. Its gradient in $\mathfrak{t}(3)$ with respect to the image-space loss $\mathcal{L}$ is

$$\frac{\partial \mathcal{L}}{\partial \mathrm{o}} = \sum_{i=1}^{K} \frac{\partial \mathcal{L}}{\partial \mathrm{p}_i}, \tag{2}$$

averaged over all pixels of the image, where $\partial \mathcal{L} / \partial \mathrm{p}_i$ is obtained from standard backpropagation through the NeRF algorithm. Similarly, the ray direction d is rigidly coupled to the camera orientation, whose gradient in $\mathfrak{so}(3)$ is stably computed as

$$\tau(\mathrm{d}) = \sum_{i=1}^{K} t_i \cdot \left(\mathrm{d} \times \frac{\partial \mathcal{L}}{\partial \mathrm{p}_i} \right), \tag{3}$$

also averaged over all pixels of the image. The direction of the vector $\tau(\mathrm{d})$ is the rotation axis of the orientation gradient and its length the magnitude. Momentum-based optimizers, such as Adam [51], must maintain their rotational moments as vectors computed from $\tau(\mathrm{d})$ and the current orientation

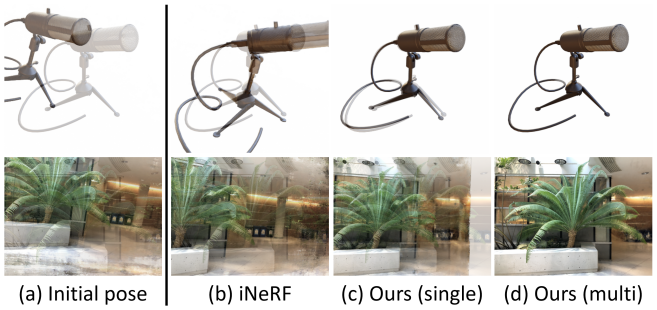

Fig. 3. A qualitative comparison between iNeRF [16] and our proposed method on both synthetic dataset [15] and real-world dataset [52], where the rendered model under the estimated pose (color) is blended with the observed image (white). We show (a) the initial pose for all the methods while (b)-(d) present the final optimized poses for different approaches.

as a 3×3 matrix whose updates are rotations induced by these vector moments.

A physical interpretation based on rigid-body mechanics is as follows: Imagine each ray-derived point as being physically attached to the camera, and the image-based loss function gradient as a force applied to that point. Then the influence is a translational force (Eq. (2)) and a torque (Eq. (3)), both acting on the camera. Hence, application of this decomposition to the Adam optimizer [51] turns Adam's first moment into physical momentum for cameras being "pushed around" by the gradients acting as forces—although Adam's second moment and exponential decay do not have straightforward physical analogues.

B. Parallelized Monte Carlo Sampling

As the loss function we optimize is non-convex over the 6-DoF space [16], it is easy for a single camera pose hypothesis to be trapped in local minima. Thanks to the computing capacity of Instant NGP [17], we are able to start the optimization from multiple hypotheses simultaneously. However, a simple multi-start idea is inefficient, especially in a large search space, where many hypotheses would be way off during the optimization process. As a result, they cannot contribute to the final optimization but still occupy a lot of computing resources.

We draw inspiration from the particle filtering framework [53]–[55] and propose a simple and effective pose hypothesis update strategy to handle this problem. We divide the optimization process into two phases, 1) free exploration and 2) resampling update. In the first phase, we generate P_N camera pose hypotheses around the start pose, with translation and rotation offsets uniformly sampled in the Euclidean space and $\mathrm{SO}(3)$, respectively. The camera pose hypotheses are optimized independently by [17] for s_1 steps. It is expected that at least some of these hypotheses will move toward the ground truth. Next, we move to the second phase, where the losses of all hypotheses are measured and taken as a reference for the sampling weight. We keep the first r ratio of the hypotheses with the lowest losses and resample the remaining hypotheses around them with small offsets. The hypotheses are optimized independently again for s_2 steps. The second phase is repeated S times, reducing r each round.

C. Pixel-based RGB Loss

One of the biggest challenges for analysis-by-synthesis pose estimation methods is that the registered model will have a different appearance than the target view, even when rendered in same pose. This issue is the result of changes in, e.g., lighting condition, occlusion, and environmental noise. While previous work such as iNeRF [16] follows a common practice to employ an L2 loss, we investigate additional losses to measure the difference between rendered and observed pixels. These loss functions vary in their treatment of visual errors as well as in their convergence properties, which in turn affect the optimization process.

Since our basic NeRF model (Instant NGP [17]) treats individual sampling rays independently, we focus exclusively on pixel-based RGB loss functions in this work: 1) *L1* is a common choice, treating errors equally. 2) *L2* penalizes larger errors more severely than smaller ones. 3) *Log L1*, as its name suggests, is a log version of the L1 loss that tries to smooth the convergence curve, especially for large errors. 4) *Relative L2* is more sensitive to cases where target pixels with high intensity are misaligned with less intense ones. 5) *MAPE* [18], or "Mean absolute percentage error", is based on the relative percentage of errors. As the L1 equivalent of the "Relative L2" loss, it is scale-independent and places heavier penalty on negative errors. 6) *sMAPE* [56], the symmetric version of MAPE, may be unstable when both the prediction and ground truth have low intensity. 7) *Smooth L1* [57] is designed to be less sensitive to outliers and can prevent exploding gradients (we set $\beta = 0.1$ empirically).

D. Implementation Details

In our experiments, the optimization process takes a total of 2560 steps ($S = 4$, so $s_1 + S s_2 = 2560$), where $P_N = 64$, $s_1 = s_2 = 512$ for the parallelized Monte Carlo sampling process. We set $r = 0.25$, which is halved each resampling round. We use the Adam optimizer [51] with learning rates that begin at 3×10^{-3} for the translation part and 5×10^{-3} for the rotation part, respectively. The learning rates decay exponentially with the base rate as 0.33 and base step as 256 over the course of optimization. The whole process takes between 15 to 20 s depending on the size of the target on a single NVIDIA RTX 3090 GPU.

V. Experimental Results

In this section, we demonstrate that our proposed method achieves improved robustness for both synthetic dataset and real-world scene compared to its predecessor iNeRF [16]. We also explore the impact of using different pixel-based RGB losses for the optimization process. These results encourage further investigation to better model the difference between the registered target and the observed view.

A. Synthetic Dataset

Setting. NeRF synthetic dataset [15] consists of 8 geometrically complex objects with no background, including

TABLE I

6-DoF POSE ESTIMATION RESULTS ON THE NERF SYNTHETIC AND LLFF DATASETS.

Method	(a) NeRF Synthetic									(b) LLFF				
	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	**Mean**	Fern	Fortress	Horns	Room	**Mean**
	Rotation error $< 5°$ (↑)									Rotation error $< 5°$ (↑)				
iNeRF [16]	0.44	0.48	0.64	0.28	0.68	0.76	0.24	0.40	0.49	0.60	0.72	0.64	0.52	0.62
Ours (single)	0.88	0.52	0.72	0.80	0.80	0.84	0.48	0.76	0.73	0.88	0.84	0.80	0.84	0.84
Ours (multiple)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.84**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	Translation error < 0.05 units (↑)									Translation error < 0.05 units (↑)				
iNeRF [16]	0.44	0.52	0.64	0.32	0.76	0.56	0.20	0.40	0.48	0.56	0.72	0.64	0.48	0.60
Ours (single)	0.88	0.52	0.72	0.76	0.80	0.56	0.28	**0.40**	0.62	0.76	0.76	0.76	0.84	0.78
Ours (multiple)	**1.00**	**1.00**	**1.00**	**0.96**	**1.00**	**1.00**	**0.84**	0.32	**0.89**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Chair, Drums, Ficus, Hotdog, Lego, Materials, Mic, and Ship. The objects have been resized to the unit box and are of complex non-Lambertian materials. Two of them (Ficus and Materials) are rendered from viewpoints sampled on a full sphere while the remaining six are rendered from viewpoints sampled on the upper hemisphere. The dataset provides camera intrinsics and extrinsics for each rendering as well as official splits for training/validation/test. All methods are trained on the training split views and evaluated on the test split for novel view pose estimation.

For each scene, we randomly choose 5 images from the test split and generate 5 different camera pose initializations. The starting pose is initialized by perturbing the ground truth pose: we rotate the camera pose around its three axes sequentially by uniformly sampling from [-15, 15] degrees, then we translate the camera along the world axes by a random offset within [-0.25, 0.25] units.

We compare our proposed method with the state-of-the-art approach iNeRF [16], where "single" and "multiple" denote our proposed without or with parallelized Monte Carlo sampling strategy, respectively. For a fair comparison, we use L2 loss for both of our method and iNeRF. For better accessibility, we use the vanilla NeRF model [15] as iNeRF's basis. We re-trained the NeRF model [58] for 200k iterations with 64 samples for its coarse network and 128 samples for the fine network while setting the batch size to 4096. At inference time, we optimized iNeRF for 500 steps following its interest resampling strategy. The sampling number and batch size are set the same as training time. In this way, we can make full use of the computing capacity and maximize its performance as noted by iNeRF [16]. The optimization process takes around 145 sec. for each object.

Results. We report the percentage of predicted poses whose error is less than 5 degrees or 0.05 units following [59]. Table Ia highlights the performance of our proposed method. We make substantial improvements over iNeRF [16] on all the objects. An example is shown in Figure 4a where the parallel Monte Carlo sampling strategy achieves better and faster evolution compared with the single pose hypothesis updates. Noticeably, some targets, e.g., Mic and Ship, are more difficult. The thin body of the Mic is challenging for the optimization when the initial rendering has small

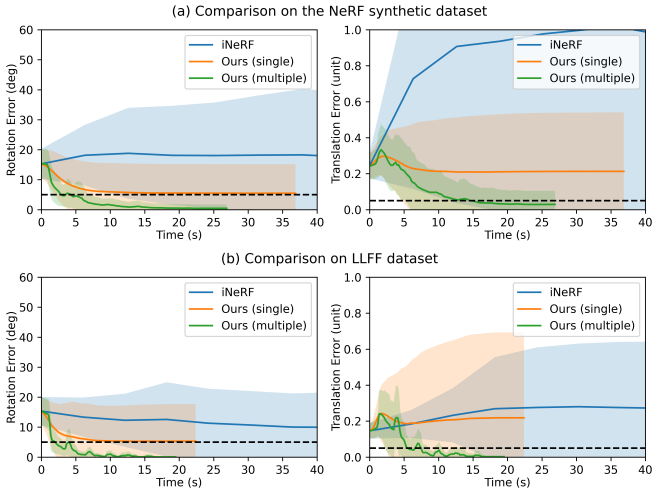

Fig. 4. Quantitative comparison of the optimization time between iNeRF [16] and our proposed method on all the scenes from both synthetic dataset [15] (top) and real-world dataset [52] (bottom). Shown are the mean (solid lines), ± 1 standard deviation (shaded areas), and the criterion (5 degrees and 0.05 units) used in our experiment (black dotted line). As shown in the plot, iNeRF is vulnerable to large start pose error, and the update oftentimes drives it away from the target. Our single hypothesis variant performs better but gets trapped in local minima. On the other hand, our proposed method with the parallelized Monte Carlo sampling module (the best camera pose hypothesis is shown) achieves more robust performance and converges to the target region faster.

overlap with the observed view, causing many samples to lie outside the object. Similarly, the large textureless areas (water) of Ship (see Fig. 5) cause sampling rays to receive a small loss when the rendered and the observed views are misaligned, an issue that becomes more severe as the number of sampling rays per hypothesis decreases under the same computation budget and fewer rays are aimed at the textured regions. We leave these challenges for future research on better importance sampling.

B. *Real-world Scene*

Setting. LLFF dataset [52] is of complex real-world scenes captured with a handheld cellphone in a roughly forward-facing manner. Different scenes have images ranging from 20 to 62, while one-eighth of them is held out for the test split. We compare our method against iNeRF [16] on the four selected scenes: Fern, Fortress, Horns, and Room. To speed up the training process of iNeRF's basic NeRF model [58],

TABLE II

ABLATION STUDY ON DIFFERENT PIXEL-BASED RGB LOSSES.

Dataset	L1	L2	Log L1	Relative L2	MAPE	sMAPE	Smooth L1
Rotation error $< 5^\circ$ (↑)							
Synthetic [15] (w/ simulated noise)	0.76	0.71	0.79	**0.86**	0.82	0.70	0.77
Real [52] (reconstruction error)	**0.89**	0.84	0.88	0.53	0.88	0.87	0.87
Mean	0.83	0.78	0.84	0.70	**0.85**	0.79	0.82
Translation error < 0.05 units (↑)							
Synthetic [15] (w/ simulated noise)	0.65	0.58	0.70	**0.78**	0.76	0.62	0.65
Real [52] (reconstruction error)	**0.87**	0.78	0.86	0.37	0.82	0.85	0.85
Mean	0.76	0.68	0.78	0.57	**0.79**	0.73	0.75

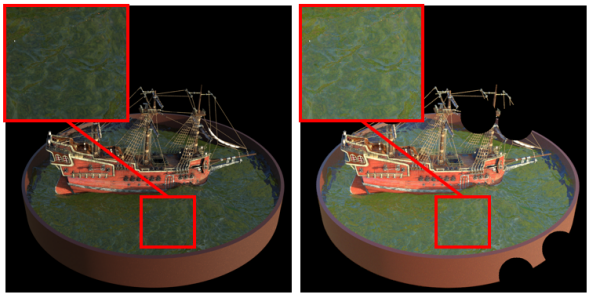

Fig. 5. Visualization of the observed view from NeRF synthetic dataset [15]. LEFT: the original test image; RIGHT: the corrupted image with simulated Gaussian & Poisson noise, brightness change, and the missing pixels. Note that the cropped region in the red bounding box (water) is textureless, making it hard to deal with.

the high resolution images are downsampled by a factor of eight before feeding into all evaluated methods.

We adopt a similar procedure to generate start poses as described in Section V-A. Considering that the views far away from the original camera center have too much artifact as all cameras are forward-facing, we change the translation perturbation range to [-0.15, 0.15] units following [16].

Results. We use the same metric in Section V-A to measure performance. The results in Table Ib demonstrate that our proposed method with a single camera hypothesis has already improved over iNeRF [16] on all the real-world scenes. It indicates that our revised gradient policy (decoupled with angular momentum) based on Adam optimizer [51] is helpful on the optimization process. The parallelized Monte Carlo sampling strategy makes it even better as it can help alleviate the issue of trapping into local minima for a single camera pose hypothesis.

C. Ablation Study on Different Pixel-based RGB Losses

Setting. In this experiment, we are interested in the robustness of different pixel-based RGB losses to various errors introduced in the procedure. Since the NeRF synthetic dataset is rendered under a perfect simulation scenario, in addition to the procedure in Section V-A, we simulate different kinds of disturbances on the test split images. They include environmental noise (Gaussian & Poisson), lighting condition difference (brightness change), and missing pixels due to potential occlusion. A sample is shown in Figure 5. The goal is to demonstrate the ability of the loss to handle potential visual difference between the rendered model (trained on the perfect simulation images) and the corrupted observed target image. Similar to Section V-B, we also evaluate different variants on the LLFF dataset [52]. As the training images and observed test image are captured in the same sequence of a specific scene, the visual difference mainly comes from the reconstruction process. We compare seven variants with different pixel-based RGB losses described in Section IV-C.

Results. As shown by the results in Table II, our proposed optimization method differ significantly depending on the loss function used. In our context, the common practice of L2 loss [15], [16] does not perform as well as other loss options. Relative L2 loss performs best on the synthetic dataset with simulated noise while it is sensitive to the reconstruction error introduced in the real dataset. On the other hand, L1 gets the best results on the real dataset but slightly worse on the synthetic dataset. Overall, MAPE achieves the best balance across two datasets, serving as a better option to deal with various errors. This finding echos the recent exploration of RawNeRF [60] to handle high dynamic range scenes with a specially designed loss function.

VI. CONCLUSION

We have proposed a parallelized optimization method based on Neural Radiance Fields (NeRF) for estimating 6-DoF poses with monocular RGB-only input. Our method performs accurate pose estimation with momentum-based camera extrinsics optimization integrated into a fast NeRF method. We have demonstrated the advantage of parallelized Monte Carlo sampling in handling local minima, and its improved efficiency in a vast search space. We have also shown the importance of a more robust pixel-based loss function for various errors. The proposed method achieves improved robustness over both synthetic and real-world datasets. While our proposed method is currently optimized for offline applications, future research will be aimed at improving speed. Additionally, a better model of the visual difference (to handle, e.g., lighting differences and occlusion) between the registered model and the observed target is needed for enhanced robustness.

ACKNOWLEDGMENTS

We thank Chen-Hsuan Lin and Yen-Chen Lin for discussions related to the work, and Rogelio Olguin for assistance with the graphics. This work was supported in part by NSF Award #2026611.

REFERENCES

[1] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] B. Wen and K. Bekris, "BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[5] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, 2018.

[6] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-Stage keypoint-based category-level object pose estimation from an RGB image," in *International Conference on Robotics and Automation (ICRA)*, 2022.

[7] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics and Science Systems (RSS)*, 2018.

[8] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *European Conference on Computer Vision (ECCV)*, 2018.

[9] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] R. L. Haugaard and A. G. Buch, "SurfEmb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] F. Manhardt, G. Wang, B. Busam, M. Nickel, S. Meier, L. Minciullo, X. Ji, and N. Navab, "CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning," *arXiv preprint arXiv:2003.05848*, 2020.

[13] T. Hou, A. Ahmadyan, L. Zhang, J. Wei, and M. Grundmann, "MobilePose: Real-time pose estimation for unseen objects with weak shape supervision," *arXiv preprint arXiv:2003.03522*, 2020.

[14] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] 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 *European Conference on Computer Vision (ECCV)*, 2020.

[16] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[17] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 41, no. 4, Jul. 2022.

[18] A. D. Myttenaere, B. Golden, B. L. Grand, and F. Rossi, "Mean absolute percentage error for regression models," in *European Symposium on Artificial Neural Networks (ESANN)*, 2015.

[19] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "MetaSDF: Meta-learning signed distance functions," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[22] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022.

[23] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[24] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "KiloNeRF: Speeding up neural radiance fields with thousands of tiny mlps," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[25] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[26] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-adjusting neural radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[27] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF−−: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[28] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[29] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3D neural scene representations for visuomotor control," in *Conference on Robot Learning (CoRL)*, 2022.

[31] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning (CoRL)*, 2022.

[32] F. Li, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, "NeRF-Pose: A first-reconstruct-then-regress approach for weakly-supervised 6D object pose estimation," *arXiv preprint arXiv:2203.04802*, 2022.

[33] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning dense object descriptors from neural radiance fields," in *International Conference on Robotics and Automation (ICRA)*, 2022.

[34] C. Xie, K. Park, R. Martin-Brualla, and M. Brown, "Fig-NeRF: Figure-ground neural radiance fields for 3D object category modelling," in *International Conference on 3D Vision*, 2021.

[35] K. Pauwels and D. Kragic, "SimTrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[36] S. Trinh, F. Spindler, E. Marchand, and F. Chaumette, "A modular framework for model-based visual tracking using edge, texture and depth features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[37] G. Pitteri, A. Bugeau, S. Ilic, and V. Lepetit, "3D object detection and pose estimation of unseen objects in color images with local surface embeddings," in *Asian Conference on Computer Vision (ACCV)*, 2020.

[38] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hands," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[39] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixel-wise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[40] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. D. Lillo, and Y. Lanthony, "AliceVision Meshroom: An open-source 3D reconstruction pipeline," in *ACM Multimedia Systems Conference*, 2021.

[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004.

[42] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*. Springer, 2006.

[43] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou, "OnePose: One-Shot object pose estimation without CAD models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[44] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[45] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $O(n)$ solution to the PnP problem," *International Journal of Computer Vision (IJCV)*, vol. 81, no. 2, 2009.
[46] I. Shugurov, F. Li, B. Busam, and S. Ilic, "OSOP: A multi-stage one shot object pose estimation framework," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[47] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang, "Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images," *arXiv preprint arXiv:2204.10776*, 2022.
[48] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018.
[49] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se (3)-Tracknet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
[50] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-NeRF: Monte Carlo localization using neural radiance fields," in *arXiv:2209.09050*, 2022.
[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
[52] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 38, no. 4, 2019.
[53] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *International Symposium on Image and Signal Processing and Analysis*, 2005.
[54] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 4, 2012.
[55] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A rao-blackwellized particle filter for 6D object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, 2021.
[56] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, 2006.
[57] R. Girshick, "Fast R-CNN," in *IEEE/CVF international Conference on Computer Vision (ICCV)*, 2015.
[58] L. Yen-Chen, "NeRF-pytorch," https://github.com/yenchenlin/nerf-pytorch/, 2020.
[59] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis *et al.*, "BOP: Benchmark for 6D object pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018.
[60] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "NeRF in the dark: High dynamic range view synthesis from noisy raw images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.