BARF 🤢: Bundle-Adjusting Neural Radiance Fields

Chen-Hsuan Lin[1] Wei-Chiu Ma[2] Antonio Torralba[2] Simon Lucey[1,3]
[1]Carnegie Mellon University [2]Massachusetts Institute of Technology [3]The University of Adelaide
https://chenhsuanlin.bitbucket.io/bundle-adjusting-NeRF

A. Visualizing the Basin of Attraction

The planar image alignment setting allows us to analyze how positional encoding affects the basin of attraction. We use the same image in Fig. 3 and consider the simpler case of aligning two image patches differing by an offset. We use a translational warp $\mathbf{p}\in\mathbb{R}^2$ on a square box whose size is $1/3$ of the raw image height and initialized to the raw center. We aim to register the center box to a single target patch of the same size shifted by some offset, shown in Fig. 1(a). We optimize the image neural network f with the objective in (4), where $\mathcal{I}_1$ is the center patch and $\mathcal{I}_2$ is the target patch, and investigate the convergence behavior of translational alignment as a function of target offsets. We search over the entire pixel grid to as far as where the target patch has no overlapping region with the initial center box.

We visualize the results in Fig. 1. Naïve positional encoding results in a more nonlinear alignment landscape and a smaller basin of attraction, while not using positional encoding sacrifices the reconstruction quality due to the limited representability of the network f. In contrast, BARF can widen the basin of attraction while reconstructing the image representation with high fidelity. This also justifies the importance of coarse-to-fine registration for NeRF in the 3D case. Please also refer to the supplementary videos for more visualizations of the basin of attraction.

B. Additional NeRF Details & Results

We provide more details and results from our NeRF experiments in this section (for real-world scenes in particular).

B.1. Evaluation Details

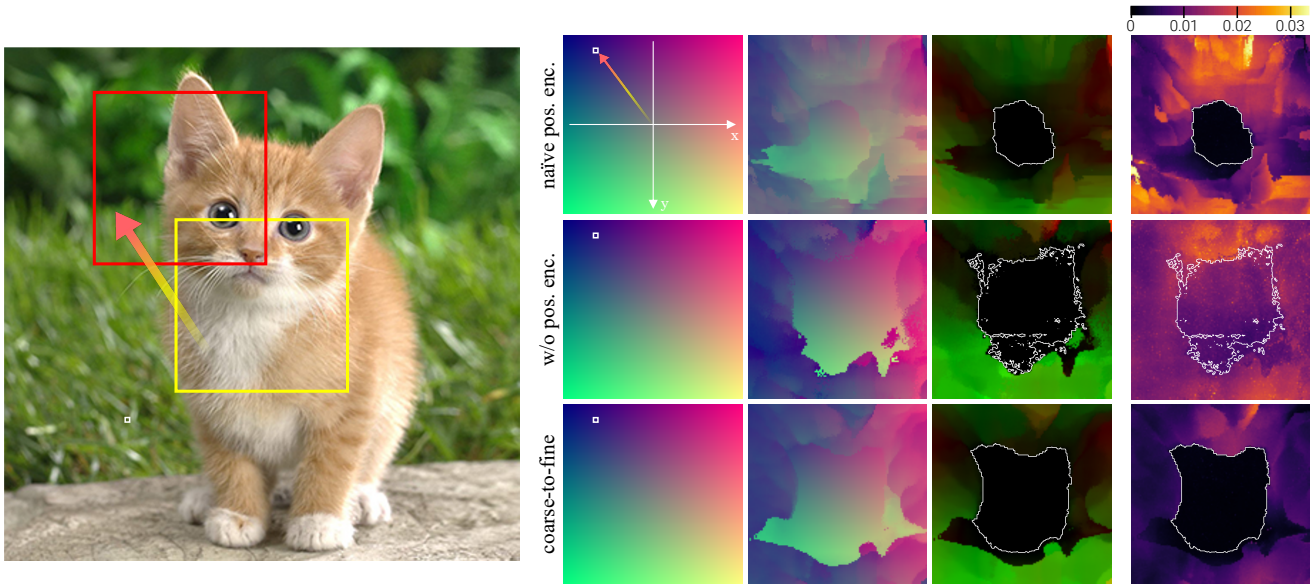
As mentioned in the main paper, the optimized solutions of the 3D scenes and camera poses are up to a 3D similarity transformation. Therefore, we evaluate the quality of registration by pre-aligning the optimized poses to the reference poses, which are the ground truth poses for the synthetic objects (Sec. 4.2) and pose estimation computed from S*f*M packages [2] for the real-world scenes (Sec. 4.3).

We use Procrustes analysis on the camera locations for aligning the coordinate systems. The algorithm details are de-

Algorithm 1: Pre-align camera poses for evaluation

Function PREALIGN($\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M,\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$):
  Input : reference poses $\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M$, optimized poses $\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$
  Output : optimized poses $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M$ aligned to the reference poses
  for $i=\{1,\dots,M\}$ **do**
    $\mathbf{o}_i=-\mathbf{R}_i^\top\mathbf{t}_i$
    $\hat{\mathbf{o}}_i=-\widehat{\mathbf{R}}_i^\top\hat{\mathbf{t}}_i$
  end
  $s,\hat{s},\mathbf{t},\hat{\mathbf{t}},\mathbf{R}=$ PROCRUSTES($\{\mathbf{o}_i\}_{i=1}^M,\{\hat{\mathbf{o}}_i\}_{i=1}^M$)
  for $i=\{1,\dots,M\}$ **do**
    $\hat{\mathbf{o}}'_i=s\mathbf{R}\left(\frac{1}{\hat{s}}(\hat{\mathbf{o}}_i-\hat{\mathbf{t}})\right)+\mathbf{t}$
    $\widehat{\mathbf{R}}'_i=\widehat{\mathbf{R}}_i\mathbf{R}^\top$
    $\hat{\mathbf{t}}'_i=-\widehat{\mathbf{R}}_i'^\top\hat{\mathbf{o}}'_i$
  end
  return $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M$
end

Function PROCRUSTES($\{\mathbf{o}_i\}_{i=1}^M,\{\hat{\mathbf{o}}_i\}_{i=1}^M$):
  Input : reference camera centers $\{\mathbf{o}_i\}_{i=1}^M$, optimized camera centers $\{\hat{\mathbf{o}}_i\}_{i=1}^M$
  Output : scale s, $\hat{s}$, translation $\mathbf{t}$, $\hat{\mathbf{t}}$, rotation $\mathbf{R}$
  $\mathbf{t}=\frac{1}{M}\sum_{i=1}^M\mathbf{o}_i\ \in\mathbb{R}^3$
  $\hat{\mathbf{t}}=\frac{1}{M}\sum_{i=1}^M\hat{\mathbf{o}}_i\ \in\mathbb{R}^3$
  $s=\sqrt{\frac{1}{M}\sum_{i=1}^M\|\mathbf{o}_i-\mathbf{t}\|_2^2}\ \in\mathbb{R}$
  $\hat{s}=\sqrt{\frac{1}{M}\sum_{i=1}^M\|\hat{\mathbf{o}}_i-\hat{\mathbf{t}}\|_2^2}\ \in\mathbb{R}$
  $\mathbf{X}=\frac{1}{s}\left([\mathbf{o}_1,\dots,\mathbf{o}_M]-\mathbf{t}\mathbf{1}_M^\top\right)\ \in\mathbb{R}^{3\times M}$
  $\widehat{\mathbf{X}}=\frac{1}{\hat{s}}\left([\hat{\mathbf{o}}_1,\dots,\hat{\mathbf{o}}_M]-\hat{\mathbf{t}}\mathbf{1}_M^\top\right)\ \in\mathbb{R}^{3\times M}$
  $\mathbf{U},\mathbf{S},\mathbf{V}^\top=\text{SVD}(\mathbf{X}\widehat{\mathbf{X}}^\top)$
  $\mathbf{R}=\mathbf{U}\mathbf{V}^\top\ \in\mathbb{R}^{3\times3}$
  if $\det(\mathbf{R})=-1$ **then**
    multiply last row of $\mathbf{R}$ by -1
  end
  return s, $\hat{s}$, $\mathbf{t}$, $\hat{\mathbf{t}}$, $\mathbf{R}$
end



(a) source and target patch (b) target warp (c) optimized warp (d) warp error (e) image recon. error

Figure 1: Visualization of the **basin of attraction**. (a) We aim to align a center box (yellow) to a target patch (red) at *every* possible location within the raw image. For each target patch, we jointly optimize f and the translational warp $\mathbf{p}$ to analyze the final warp error and the image reconstruction loss. (b) The target offsets forms a color-coded map, where green indicates horizontal offsets and red indicates vertical offsets. The above example corresponds to the highlighted pixel. (c) The optimized warp parameters and (d) the warp error for every target patch location, where the white contours highlight the offset error threshold of 0.5 pixels. BARF effectively widens the basin of attraction (range of successful alignment) with a smoother landscape compared to naïve positional encoding. (e) Without positional encoding, f has limited capacity of representing the image details, resulting in nonzero image errors despite the registration being successful as well.

scribed in Alg. 1. We write the reference poses $\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M$ and the optimized poses $\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$ in the form of camera extrinsic matrices, and the aligned poses can be written as $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M = \textsc{PreAlign}(\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M,\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M)$. After the cameras are Procrustes-aligned, we apply the relative rotation (solved for via the Procrustes analysis process) to account for rotational differences. We measure the rotation error between the SfM poses and the aligned poses from NeRF/BARF by the angular distance as

$$\Delta\theta_i = \cos^{-1}\frac{\text{trace}\big(\mathbf{R}_i\widehat{\mathbf{R}}_i^{\prime\top}\big)-1}{2}\ ,\quad i=\{1,\dots,M\}\ , \tag{1}$$

where $\langle\cdot,\cdot\rangle$ is the quaternion inner product. For additional clarity, we provide a more detailed visualization of the optimized camera poses in Fig. 2 (for the LLFF dataset).

To evaluate the quality of novel view synthesis while being minimally affected by camera misalignment, we transform the test views (provided by Mildenhall *et al*. [1]) to the coordinate system of the optimized poses by applying the scale/rotation/translation from the Procrustes analysis, as in Alg. 1. The camera trajectories from the baseline NeRF with naïve full positional encoding exhibits large rotational and translational differences compared to SfM poses in general. For this reason, the view synthesis results from the baseline NeRF, whose corresponding test views are also determined using Procrustes analysis, are far from plausible. Unfortunately, there is no other systematic way of determining what the corresponding views held out from the SfM poses would be in the learned coordinate system. Nevertheless, we provide additional qualitative results in Fig. 3, where the novel views are selected from a *training* view closest to the average pose and sampling translational perturbations. Please also see the supplementary video for more details.

B.2. Real-World Scenes (LLFF Dataset)

Dataset. The LLFF dataset [1] consists of 8 forward-facing scenes with RGB images sequentially captured by hand-held cameras. In the original NeRF paper [1], the test views were selected by holding out every 8th frame from the video sequence and training with the remaining frames. Unlike Mildenhall *et al*. [1], however, we hold out the last 10% of the frames for evaluation and train with the first 90% frames. This train/test split does not assume that the held-out views are interpolations of the training views, which allows a more practical simulation of predicting future viewpoints from

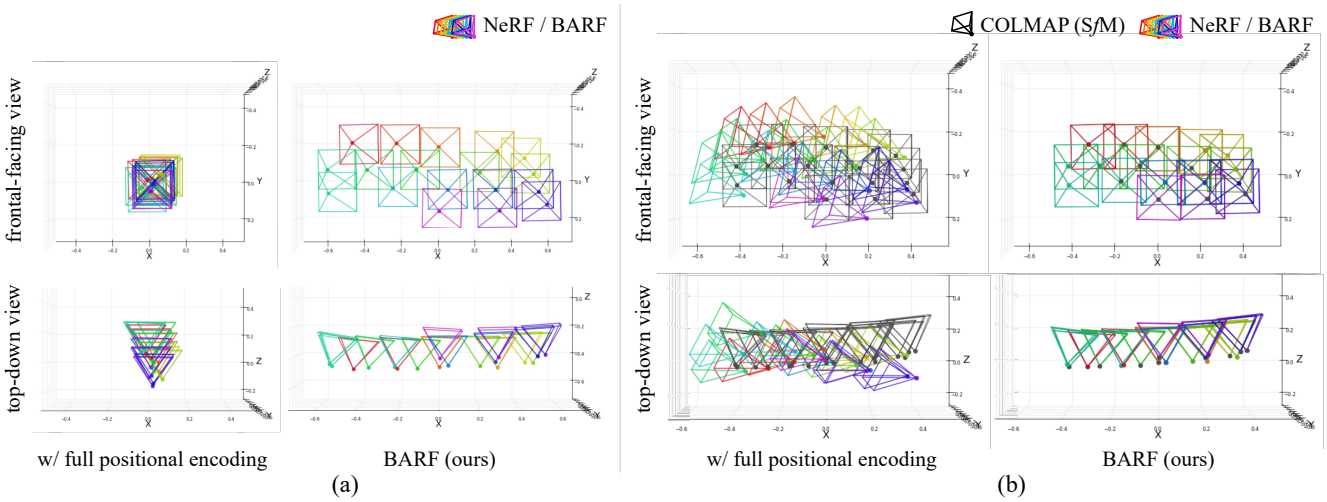


Figure 2: Visualization of the **optimized camera poses** for the *fern* scene. The poses for both the baseline NeRF (with full positional encoding) and BARF are initialized to the identity transform for all frames. (a) The camera poses of the baseline NeRF get stuck in a suboptimal solution that does not accurately reflect the actual viewpoints, whereas BARF can effectively optimize for the underlying poses. (b) We compare the optimized poses to those computed from S*f*M [2] (colored in black), where we align the pose trajectories using Procrustes analysis. The camera poses optimized by BARF highly agree with those from S*f*M, whereas those from the baseline NeRF cannot be well-aligned with Procrustes analysis. Therefore, there is no systematic way of finding a reasonable set of corresponding held-out views with respect to the optimized coordinate system.

	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-rex
split	18 / 2	31 / 3	38 / 4	56 / 6	24 / 2	23 / 2	37 / 4	50 / 5
total	20	34	42	62	26	25	41	55

Table 1: Dataset statistics of the train/test splits for the real-world scene (LLFF) experiments, where we hold out the last 10% frames from each sequences.

previous observations. The statistics of the train/test split for each scene is provided in Table 1.

Full comparison. We provide a more complete evaluation of the LLFF experiment in Table 2, where we also include the baseline without any positional encoding. Note that we consider the same schedule for all scenes in the dataset (adjusting the positional encoding from iterations 20K to 100K); due to the per-scene optimization nature, however, the optimal coarse-to-fine scheduling for each scene would actually be data-dependent. Despite this, the coarse-to-fine scheduling considered here already allows BARF to achieve an averaged similar or better performance on real-world scenes. An exhaustive analysis of searching for the best scheduling is currently out of scope of this paper.

In the main LLFF experiments, we sample 3D points along each ray linearly in the inverse depth (disparity) space, where the lower and upper bounds are the image plane and infinity respectively (*i.e.* $1/z_{\text{near}} = 1$ and $1/z_{\text{far}} = 0$). To analyze the effect of depth parametrization on the performance of real-world scenes, we run an additional set of the same experiments by sampling the 3D points in the regular (metric) depth space, bounded by $z_{\text{near}} = 1$ and $z_{\text{far}} = 20$.

We report the quantitative results in Table 3. The baseline NeRF with full positional encoding still performs poorly in all metrics. Although the baseline without positional encoding may be slightly better than BARF in this setup, all methods being compared here exhibit better performance when the 3D points are sampled in the inverse depth space. We present empirical results as a supplement and leave a complete analysis of depth parametrization to future work.

References

[1] 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. 2

[2] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 3

Figure 3: Additional novel view synthesis results from the real-world scene experiment (LLFF dataset). Instead of visualizing the held-out views computed by Procrustes analysis, we show qualitative results at new viewpoints by sampling camera pose perturbations around the viewpoint from the training set (closest to the average pose). Note that for this set of qualitative results, we do not have ground-truth RGB images to compare against. BARF can optimize for scene representations of much higher quality. Please refer to the supplementary video for more details.

Scene	Camera pose registration						View synthesis quality											
	Rotation (°) ↓			Translation ↓			PSNR ↑				SSIM ↑				LPIPS ↓			
	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF
Fern	74.452	0.194	**0.191**	30.167	0.194	**0.192**	9.81	23.73	**23.79**	23.72	0.187	0.709	**0.710**	0.733	0.853	0.371	**0.311**	0.262
Flower	2.525	0.883	**0.251**	2.635	0.297	**0.224**	17.08	**24.66**	23.37	23.24	0.344	**0.739**	0.698	0.668	0.490	**0.200**	0.211	0.244
Fortress	75.094	**0.320**	0.479	33.231	**0.289**	0.364	12.15	28.35	**29.08**	25.97	0.270	0.774	**0.823**	0.786	0.807	0.206	**0.132**	0.185
Horns	58.764	**0.182**	0.304	32.664	**0.170**	0.222	8.89	22.27	**22.78**	20.35	0.158	0.724	**0.727**	0.624	0.805	0.312	**0.298**	0.421
Leaves	88.091	2.938	**1.272**	13.540	0.468	**0.249**	9.64	**19.08**	18.78	15.33	0.067	**0.566**	0.537	0.306	0.782	0.375	**0.353**	0.526
Orchids	37.104	**0.550**	0.627	20.312	**0.396**	0.404	9.42	19.27	**19.45**	17.34	0.085	0.566	**0.574**	0.518	0.806	0.313	**0.291**	0.307
Room	173.811	0.384	**0.320**	66.922	0.311	**0.270**	10.78	30.71	**31.95**	32.42	0.278	0.928	**0.940**	0.948	0.871	0.135	**0.099**	0.080
T-rex	166.231	**0.138**	1.138	53.309	**0.261**	0.720	10.48	22.48	**22.55**	22.12	0.158	**0.783**	0.767	0.739	0.885	**0.197**	0.206	0.244
Mean	84.509	0.699	**0.573**	31.598	**0.298**	0.331	11.03	23.82	**23.97**	22.56	0.193	**0.724**	0.722	0.665	0.787	0.264	**0.238**	0.283

Table 2: Full quantitative comparison of NeRF on the LLFF forward-facing scenes from *unknown* camera poses. BARF and our baseline without positional encoding are competitive in different metrics. An optimal coarse-to-fine schedule for BARF could be theoretically found per scene that are at least as good as the baseline methods; exhaustively or adaptively search for such optimal schedule is currently out of scope of this paper. Translation errors are scaled by 100.

Scene	Camera pose registration						View synthesis quality											
	Rotation (°) ↓			Translation ↓			PSNR ↑				SSIM ↑				LPIPS ↓			
	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF
Fern	164.243	**0.391**	0.448	18.265	**0.260**	0.283	9.17	23.39	**23.55**	22.76	0.148	**0.700**	**0.700**	0.655	1.041	0.362	**0.335**	0.397
Flower	7.462	**0.177**	3.282	1.959	**0.211**	0.724	18.81	**23.63**	22.99	23.37	0.408	**0.710**	0.651	0.654	0.657	**0.224**	0.227	0.272
Fortress	172.581	**0.502**	0.576	46.673	**0.466**	0.468	11.17	26.75	**26.92**	25.67	0.222	0.684	**0.716**	0.662	1.122	0.348	**0.270**	0.403
Horns	34.840	**0.248**	0.266	18.207	**0.223**	0.228	8.95	21.52	**21.79**	20.37	0.174	**0.714**	0.701	0.599	1.028	0.325	**0.310**	0.464
Leaves	4.708	**1.194**	1.832	1.105	**0.261**	0.367	11.66	**18.36**	17.68	16.34	0.104	**0.516**	0.473	0.353	0.822	0.407	**0.356**	0.534
Orchids	172.600	0.531	**0.443**	37.887	**0.413**	**0.413**	8.22	**18.84**	18.57	16.97	0.062	**0.536**	0.513	0.402	1.086	**0.357**	0.373	0.564
Room	160.757	0.456	**0.207**	51.988	0.454	**0.203**	8.09	30.90	**31.99**	32.10	0.127	0.924	**0.938**	0.935	1.215	0.139	**0.104**	0.109
T-rex	175.893	**0.334**	5.586	61.026	**0.328**	3.085	8.30	**22.74**	21.24	22.42	0.123	**0.794**	0.731	0.770	1.174	**0.187**	0.225	0.205
Mean	111.635	**0.479**	1.580	29.639	**0.327**	0.721	10.54	**23.26**	23.09	22.50	0.171	**0.698**	0.678	0.629	1.018	0.294	**0.275**	0.368

Table 3: Quantitative results of NeRF on the LLFF forward-facing scenes from *unknown* camera poses, sampling the 3D points in the regular depth space (instead of the inverse depth space). Translation errors are scaled by 100.