Reproducibility Blogpost

BARF🤮: Bundle-Adjusting Neural Radiance Fields

paper: https://arxiv.org/pdf/2104.06405.pdf
our github: https://github.com/mateicristea88/bundle-adjusting-NeRF-DeepLearning

Dan Andreescu & Matei Cristea-Enache

In this blog post, we describe our approach at understanding BARF, reproducing some of the results presented in the original paper, and exploring possible variations on the parameters and heuristics chosen by the authors.

NERF

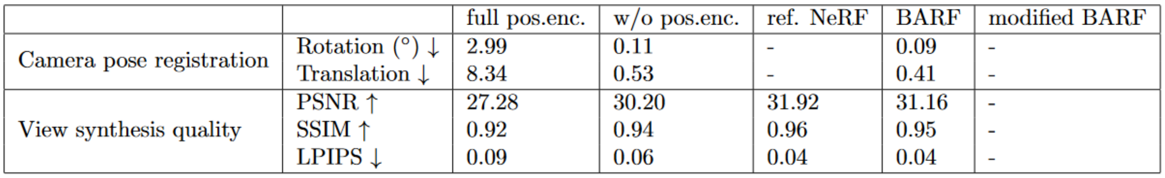
NERF (neural radiance fields) is novel approach at 3D reconstruction, that is learning a scene representation from a set of images. To that end, a neural network is trained to map 3D points in space, to color and volume density. New views of the scene can then be synthesized by shooting rays from the desired new point of view and evaluating the network. This technique has shown impressive results even on little or low resolution data. However, there is one hard requirement, and at the same time, a great limitation: the camera position of each image must be known with high accuracy in order for the network to converge to a valid scene representation.

BARF

BARF stands for bundle adjusted neural radiance fields and builds on top of NERF specifically to mitigate the aforementioned limitation. BARF allows for imperfect (or even missing) camera positions, and aims to learn the correct positions at the same time it is learning the scene representation. To understand why BARF can do that, one should first look at why a trivial extension of NERF would fail. Since estimating the camera position depends on the assumed 3D model, and estimating the 3D model depends on the assumed camera position, the authors humorously refer to this cyclic dependency as the chicken-and-egg problem. However, its consequences are nowhere near funny, as noise in the camera position will result in a suboptimal scene representation, which can result in even worse camera position estimates, and so on. BARF mitigates this destabilizing effect by facilitating the convergence of the camera positions in the earlier stages. This can be achieved by filtering away the signal components for which small perturbations lead to large differences in camera registration. Concretely, BARF applies a smooth mask on the positional encoding that filters out frequencies above a threshold. Initially, this threshold starts at zero, and gradually increases to the maximum frequency. The lower the threshold, the smoother the signals, so the more coherently they can be aligned. This makes camera position estimates less precise, as information is discarded, but more robust to noise. This effectively 'attracts' the initial estimates towards their correct positions more strongly but less precisely in the beginning so that they don't diverge away, and the other way around towards the end, so that the final estimates have high fidelity.

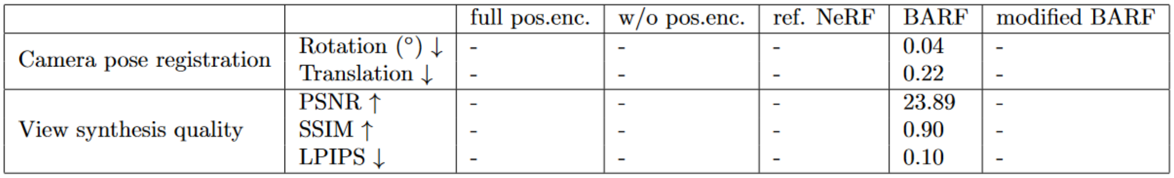
REPRODUCING

Every pice of knowledge that we share with each other and use to make decisions, develop tools, or build on top of it, would be almost worthless without trust, because we can rarely afford to thoroughly validate every bit of information we use. That is why we have to build and maintain trust in the scientific community, and there is probably no better way to do it than to check each other's results, and be vocal about the outcomes. This is why we present in this blog post the results that we were able to reproduce from the BARF paper. Due to computational constraints, we could only run a subset of the experiments, but nevertheless they all appeared very similar to the results presented in the paper (Table 2).



		full pos.enc.	w/o pos.enc.	ref. NeRF	BARF	modified BARF
Camera pose registration	Rotation (°) ↓	2.99	0.11	-	0.09	-
	Translation ↓	8.34	0.53	-	0.41	-
View synthesis quality	PSNR ↑	27.28	30.20	31.92	31.16	-
	SSIM ↑	0.92	0.94	0.96	0.95	-
	LPIPS ↓	0.09	0.06	0.04	0.04	-

Table 1: Quantitative results of NeRF and BARF on synthetic scene **chair**. Translation errors are scaled by 100.



		full pos.enc.	w/o pos.enc.	ref. NeRF	BARF	modified BARF
Camera pose registration	Rotation (°) ↓	-	-	-	0.04	-
	Translation ↓	-	-	-	0.22	-
View synthesis quality	PSNR ↑	-	-	-	23.89	-
	SSIM ↑	-	-	-	0.90	-
	LPIPS ↓	-	-	-	0.10	-

Table 2: Quantitative results of NeRF and BARF on synthetic scene **drums**. Translation errors are scaled by 100.

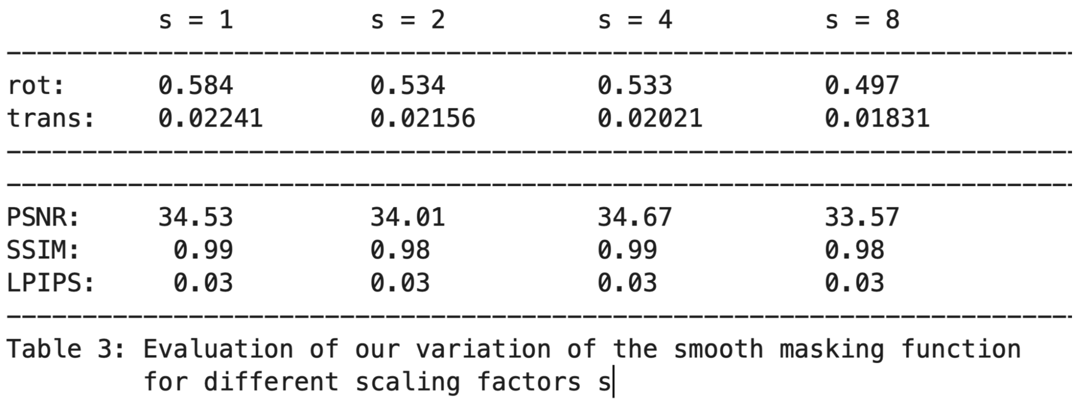
NEW VARIANT

After figuring out what the essence of this novel technique was, we wanted to experiment with variations of it. We first noted the desired behavior of the masking function - it smoothly maps the optimization progress to the a weight on the unit interval, and it is non-decreasing over the whole domain. We also noticed that only a small part of the domain maps to values that are not 0 or 1, so the transition from filtering to not filtering happens within a fraction of 1 / L of the optimization progress (where L is the number of frequency bases). As we have not found, nor could come up ourselves with a motivation for this effect, we have experimented with widening the transition window. The variant of the function now includes a scale factor s, indicating how much to scale the transition window, such that it becomes a fraction of s / L of the progress. Note that for s = 1, we get the original function (Eq. 14) in the paper.

$$w\_k(a) \begin{array}{ll} = 0 & \text{if } a < k \\ = (1 - \cos((a - k) * \pi / s)) / 2 & \text{if } 0 <= a - k < s \\ = 1 & \text{if } a - k >= s \end{array}$$

Eq. 1) our variation of smooth masking function with scaling of the transition window

We evaluated the five metrics from table 2 in the paper for four values of s <- {1,2,4,8}, but on a decreased resolution of 100x100, and shorter training iterations of 20K. While the three scores of view synthesis quality appear not to be influenced by the change, we observe a slight decreasing trend in the two error metrics corresponding to camera pose registration. Both the rotation error and the translation error appear to go down with the increase in transition window, suggesting a possibly promising direction of improvement for BARF.



	s = 1	s = 2	s = 4	s = 8
rot:	0.584	0.534	0.533	0.497
trans:	0.02241	0.02156	0.02021	0.01831
PSNR:	34.53	34.01	34.67	33.57
SSIM:	0.99	0.98	0.99	0.98
LPIPS:	0.03	0.03	0.03	0.03

Table 3: Evaluation of our variation of the smooth masking function for different scaling factors s

WORK DIVISION

Matei was responsible for reproducing results that were presented in Table 2 in the paper. Besides this criterion, he did the first experimental setup, i.e. got the code to run on Google Cloud, and took the initiative in finance and administration to arrange for necessary extra computational power. He was also the one to assume the lead communication role with the course staff.

Dan was responsible for the new algorithm variant and its evaluation, i.e. the scaling of the transition window in the smooth masking function. Besides this criterion, he pursued alternative directions of experimental setup, but could not deliver a better alternative. He was also the one to write this blog post.