A Study of Nope-NeRF for Novel View Synthesis of Handheld Objects
Image Processing and Computer Vision Project Work (3 CFU)

Ilenia Carboni
Master's Degree in Artificial Intelligence, University of Bologna
ilenia.carboni3@studio.unibo.it

Abstract

I introduce a variation of NoPe-NeRF [1] that can leverage the use of masks to perform unposed NeRF training of handheld objects without taking into account the backgrounds of the images, building on a previous proof of concept which was discussed as Bachelor's Thesis in 2022 [2]. As a further improvement, the segmentation pipeline to obtain the masks, previously obtained by mixing hand-crafted elements of classical Computer Vision with Neural Networks for Object Detection, is simplified by leveraging the recent progress in Foundation models for Computer Vision, namely the Segment Anything project [3]. As NoPe-NeRF shows an overall lower performance, paired with $\tilde{4}$x more training time and strong struggles in pose estimations, the possible use of background for contextual cues in pose and focal estimation comes into question.

1 Introduction

With the advent of wearable technology that allows high-quality image acquisitions, like the new Ray-Ban Meta Smart Glasses - formerly known as Ray-Ban Stories - there has been a renewed interest in the possibility of acquiring sparse views of an object and obtain a full 3D reconstruction, which has ideal applications in Augmented Reality and Metaverse scenarios.

An initial proof of concept was explored in my Bachelor's Thesis[2], which indeed proved that it is possible to acquire simple videos of a handheld object in a mostly unconstrained scenario and use it to obtain a novel representation using Neural Radiance Fields (NeRF)[4]. However, the work presented two main bottlenecks:

- the pipeline to obtain masks for the objects starting from the frames obtained from the videos made use of many hand-crafted steps, making it complex to replicate and all-around slow;
- as NeRF requires in principle camera poses for the input images, COLMAP [7][6] is required to generate pose estimations. This requirement proved to be a massive bottleneck, as the repeated failures to reconstruct most objects made the dataset overall unusable with few notable exceptions (the 'greendino' set).

This project work aimed to address those issues by: a) introducing a novel strategy to obtain segmentation masks, using recent innovation in Foundation Models and Large Language Models for Computer Vision, namely the Segment Anything Model (SAM) [3]; and b) attempt an unposed-NeRF training as proposed in NoPe-NeRF [1], and enable the use of masks to guide the ray-marching algorithm.

While the introduction of SAM increases the number of images at my disposal, with a theoretical 100% masks obtained, NoPe-NeRF proves to be only mildly capable of rendering known images, while massively struggling with consistency between different views, as it struggles with pose estimation without environmental cues. As joint optimization of NeRF and camera poses proves to be both unsuccessful and time-consuming, with a single run taking over two days, the question of the importance of background information arises. Having no environment information at all proved to be not beneficial, but the same level of confusion may be obtained using the original background; as the images are acquired by making the object rotate in front of the glasses, it is overall static. Further works may try and address the possibility of keeping the background to add context.

Code can be found at https://github.com/ileniacarboni/masked-nope-nerf.

2 Methods

NeRF[4] algorithms start from a particular viewpoint to march rays through a scene, obtaining a sampled set of 3D points, along with the corresponding 2D view directions; the resulting 5D coordinates are used as inputs to a neural network, a Multi-Layer Perceptron (MLP), to produce an output set of colors and densities. The 4D result is then rendered into a 2D image using classical techniques.

As not all the areas of an image are informative, this work addresses how to concentrate rays in areas figuring only the object of interest. This is obtained by adding another input to the network: a binary segmentation mask of the object. This segmentation mask is used to attribute each pixel of the image a probability: $1.$ if the pixel belongs to the object, $0.$ otherwise. In practice, as

background can in general provide cues, the latter value is set to a small, non-zero quantity.

To obtain valid segmentation masks, my previous work [2] employed an overall handcrafted pipeline that included steps from both classical computer vision and neural networks for hand segmentation and object detection, giving in output either a mask or nothing. To refine the process and remove most human intervention, I leverage the advent of Foundation Models for Computer Vision using Segment Anything Model (SAM) [3]. SAM is a promptable, zero-shot segmentation system from Meta Research that allows to generate masks starting from a variety of prompts like points or object bounding boxes from an object detection task. It can also automatically generate masks for the full image, thanks to its general notion of what is an object.

The resulting masks are used to remove the background on the dataset. They are also used as further input to the network.

I do not modify the behavior of NoPe-NeRF. Both pose and focal estimation networks are left untouched. My contributions are:

- Accept as input and process binary segmentation masks
- March only rays passing through the object using random sampling on the image, weighted by the value of respective pixels in the segmentation mask

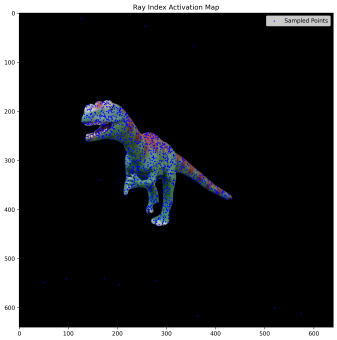


Figure 1: Graphic Illustration of the modified ray marching algorithm, using 1024 points (same as the code used)

3 Experiments

3.1 Segmentation Masks Generation

Different strategies were tested to maximize the number of correct segmentation masks, while minimizing the need for successive hand-made removal. While prompting the network with coordinate ranges for the objects showed an overall success, the final chosen strategy was to:

- Generate all the possible masks for each input image. This produced as output a series of folders containing all masks, in order from the most to the least confident, and a metadata CSV file.
- For each folder, use the metadata CSV file and a series of information on general location of the object in the scene - overall in the middle section of the picture, but not always in the center - to obtain the complying mask with the highest confidence.
- Remove any spurious mask by hand.

With this process, I am able to obtain masks for each set, while the original work lacked results for the 'book' set, overall improving or matching previous results; while the automatic cleanup can in some cases introduce errors (we are only able to recover 77% of the masks), generating all masks allows to have the option to hand-select the correct mask if desired. This means that theoretically, we can achieve the objective of 100% success rate in segmentation, compared to the maximum achievable with the original pipeline (75%).

In the following sections, I will work only with the masked 'greendino' set and without introducing the missing masks (for a total of 91% usable images from the set). This set, the most successful from the previous work, is also one of the only three benchmarked with the original NeRF algorithm, meaning I can perform a comparative study.

For memory purposes, the set is downscaled to 640x640.

3.2 NoPe-NeRF Experiment

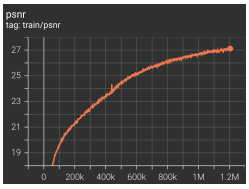
The modified NoPe-NeRF algorithm is trained for 5500 epochs on a NVIDIA GeForce GTX 1080 Ti, about a quarter of the original experiment runtime, for a total of 1.2M total iterations. While I reduced the number of total epochs, I also changed the scheduling setup, with the scheduler running right from the first epoch instead of after the 50% mark. The total run time amounted to 50 hours and 36 minutes, about four times more than the original NeRF [4].

	PSNR (Training)	PSNR (Holdout)	Time
NeRF	34.22	26.99	$\sim$12h
NoPe-NeRF	27.116	23.08	50h 36m

Table 1: Final Results of the comparison

Results in 1 show worse results than the original work. While the network achieves good levels of rendering on the training set, smoother than the original work, the network struggles to reconstruct the object and generate consistent views. The issue becomes more evident when the generated depth maps are observed, showing that the network struggles to place the object and is plagued by artifacts.

This is actually in line with the absence of results on common benchmark datasets like the NeRF Synthetic 360° [4], as the authors declared their method was unsuited for synthetic scenes. Indeed, background removal, which aided reconstruction in NeRF along with



(a) PSNR (Rendering)/steps

(b) End of training Evaluation

Figure 2: Training Results for NoPe-NeRF

focused ray marching, might have overcomplicated the task of pose learning for NoPe-NeRF, due to the absence of environmental cues. While it is possible to use the full original images, it should be noted that the background remains static throughout the set; while it would probably bring more contextual information, it might also contribute to further confuse the network.

4 Conclusions

The project introduces a variation of NoPe-NeRF [1] that is capable of using binary segmentation masks as an additional input to perform unposed NeRF training of handheld objects without taking into account the backgrounds of the images.

The new process to obtain segmentation masks widely improves the final quality, with theoretically no image lost in the process and a full set recovered with respect to the original work. On the other side, the use of NoPe-NeRF on the masked images shows an overall lower performance in a vastly higher amount of time, struggling with pose estimation and consistency between views. This raises the issue of contextual information for pose estimation, which seems to be the weak link in the examined dataset.

Further work may move in the direction of testing the setup on the full images, to see if the introduction of more cues improves the overall results; another possible solution would be to test on different, higher-quality datasets of handheld objects.

References

[1] Wenjing Bian et al. “NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior”. In: 2023.

[2] Ilenia Carboni. “Acquisizione di un dataset tramite smart glasses per l’addestramento di Neural Radiance Fields”. Final Project for Bachelor’s Degree in Computer Engineering, A.Y. 2021/2022.

[3] Alexander Kirillov et al. “Segment Anything”. In: *arXiv:2304.02643* (2023).

[4] Ben Mildenhall et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. In: *ECCV*. 2020.

[5] Thomas Müller et al. “Instant Neural Graphics Primitives with a Multiresolution Hash Encoding”. In: *ACM Trans. Graph.* 41.4 (July 2022), 102:1–102:15. DOI: 10.1145/3528223.3530127. URL: https://doi.org/10.1145/3528223.3530127.

[6] Johannes Lutz Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[7] Johannes Lutz Schönberger et al. “Pixelwise View Selection for Unstructured Multi-View Stereo”. In: *European Conference on Computer Vision (ECCV)*. 2016.

Appendix

A Results on COLMAP and Instant-NGP

To fully assess the extent of improvement the Segment Anything Model (SAM) [3] allows, I show a further experiment to evaluate the improvement on NeRF without the added complexity of simultaneous Pose Estimation, similarly to [2].

I added the missing segmentation masks, obtaining an employable full set. Then, I ran COLMAP [7][6] to obtain camera poses. Results still show an overall inability to reconstruct a full 360° view for all objects, except ’greendino’, ’yellowdino’ and ’waterbottle’, as was previously observed. However, it was possible to reconstruct at least a forward-facing view, showing that having access to more images improves the overall quality.

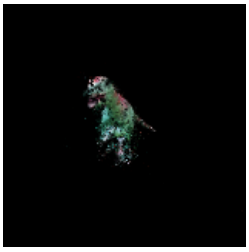
The following part is performed with the ’greendino’ 1264x1264 set, converted to RGBA format using masks. I chose Instant-NGP[5], as I had a functioning setup already and it had a much faster execution time compared to other methods. The choice of converting the images to RGBA, which was useless for the other methods, in this particular instance helps the model; while in the other methods the masks were used to march a fixed number of rays to occupied sections, Instant-NGP generates a dynamic number of rays based on previous steps’ results and only samples from occupied space, incrementally building an occupancy map.

Instant-NGP is ran for a total of 150K steps on 219 images, with the remaining 24 constituting the test set on which results are computed. All hyperparameters are left as in the original implementation. Because of the overall fast execution (each run took less than an hour), multiple tests have been performed.

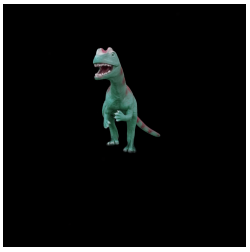
	PSNR (Holdout)	**N°Imgs**	**Time**
NeRF	26.99	205	∼12h
Instant-NGP	30.3	243	<1h

Table 2: Instant-NGP results compared to NeRF

Instant-NGP widely surpasses NeRF performance, as predicted.



(a) Rendering from the Original NeRF



(b) Rendering from Instant-NGP

Figure 3: Result comparison between Instant-NGP and NeRF

Conclusions

This appendix wanted to show that the improvement on the segmentation task alone is enough to push COLMAP to a point where it is able to partially reconstruct the object with enough fidelity, to be able to also move towards real-time reconstruction which, to my knowledge, has only been addressed by methods that required poses in input.