Creating Neural Search and Rescue Fly-Through Environments with Mega-NeRF

Originally published at https://www.unite.ai/creating-neural-search-and-rescue-fly-through-environments-with-mega-nerf/

A new research collaboration between Carnegie Mellon and autonomous driving technology company Argo AI has developed an economical method for generating dynamic fly-through environments based on Neural Radiance Fields (NeRF), using footage captured by drones.

CLICK TO VIEW ANIMATED GIF

The new approach, called Mega-NeRF, obtains a 40x speed-up compared to the average Neural Radiance Fields rendering standard, as well as offering something notably different from the standard tanks and temples that recur in new NeRF papers.

The new paper is titled *Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs*, and comes from three researchers at Carnegie Mellon, one of whom also represents Argo AI.

Modeling NeRF Landscape for Search and Rescue

The authors consider that search-and-rescue (SAR) is a likely optimal use case for their technique. When evaluating an SAR landscape, drones are currently constrained both by bandwidth and battery life restrictions, and are therefore not usually able to obtain detailed or comprehensive coverage before needing to return to base, at which point their collected data is converted to static 2D aerial view maps.

The authors state:

'We imagine a future in which neural rendering lifts this analysis into 3D, enabling response teams to inspect the field as if they were flying a drone in real-time at a level of detail far beyond the achievable with classic Structure-from-Motion (SfM).'

Tasked with this use-case, the authors have sought to create a complex NeRF-based model that can be trained inside of a day, given that the life-expectancy of survivors in search-and-rescue operations decreases by up to 80% during the first 24 hours.

The authors note that the drone capture datasets necessary to train a Mega-NeRF model are 'orders of magnitude' larger than a standard dataset for NeRF, and that model capacity must be notably higher than in a default fork or derivative of NeRF. Additionally, interactivity and explorability is essential in a search and rescue terrain map, whereas standard real-time NeRF renders are expecting a much more limited range of pre-calculated possible movement.

Divide and Conquer

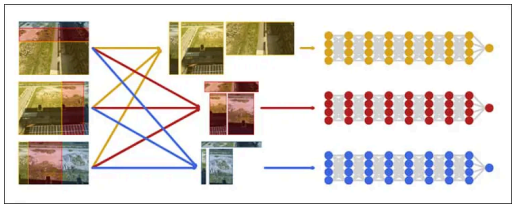
To address these issues the authors created a geometric clustering algorithm that divides the task up into submodules, and effectively creates a matrix of sub-NeRFs that are trained contemporaneously.

At the point of rendering, the authors also implement a just-in-time visualization algorithm that is responsive enough to facilitate full interactivity without excessive pre-processing, similar to the way that video games will ramp up detail on items as they approach the user's viewpoint, but which remain at an energy-saving and more rudimentary scale when in the distance.

These economies, the authors contend, lead to better detail than previous methods that attempt to address very wide subject areas in an interactive context. In terms of extrapolating detail from limited resolution video footage, the authors also note Mega-NeRF's visual improvement over the equivalent functionality in UC Berkeley's PlenOctrees.

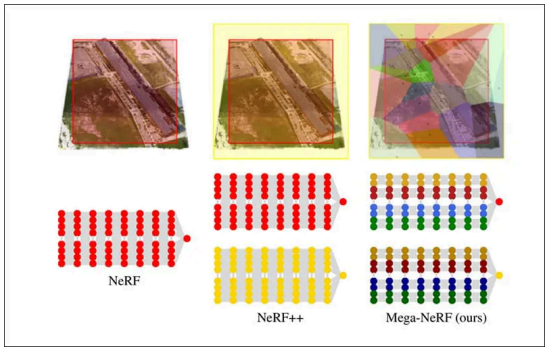
CLICK TO VIEW ANIMATED GIF

The project's use of chained sub-NeRFs is based on KiloNeRF's real-time rendering capabilities, the authors acknowledge. However, Mega-NeRF departs from this approach by actually performing 'sharding' (discrete shunting of facets of a scene) during training, rather than KiloNeRF's post-processing approach, which takes an already-calculated NeRF scene and subsequently transforms it into an explorable space.



A discrete training set is created for submodules, comprised of training image pixels whose trajectory might span the cell that it represents. Consequently, each module is trained entirely separately from adjacent cells. Source: https://arxiv.org/pdf/2112.10703.pdf

The authors characterize Mega-NeRF as *'a reformulation of the NeRF architecture that sparsifies layer connections in a spatially-aware manner, facilitating efficiency improvements at training and rendering time'*.



Conceptual comparison of training and data discretization in NeRF, NeRF++, and Mega-NeRF. Source: https://meganerf.cmusatyalab.org/

The authors claim that Mega-NeRF's use of novel temporal coherence strategies avoids the need for excessive pre-processing, overcomes intrinsic limits on scale, and enacts a higher level of detail than prior similar works, without sacrificing interactivity, or necessitating multiple days of training.

The researchers are also making available large-scale datasets containing thousands of high-definition images obtained from drone footage captured over 100,000 square meters of land around an industrial complex. The two available datasets are 'Building' and 'Rubble'.

Improving on Prior Work

The paper notes that previous efforts in a similar vein, including SneRG, PlenOctree, and FastNeRF, all rely on some kind of caching or pre-processing that adds compute and/or time overheads that are unsuitable for the creation of virtual search-and-rescue environments.

While KiloNeRF derives sub-NeRFs from an existing collection of multilayer perceptrons (MLPs), it is architecturally constrained to interior scenes with limited extensibility or capacity to address higher-scale environments. FastNeRF, meanwhile, stores a 'baked', pre-computed version of the NeRF model into a dedicated data structure and allows the end-user to navigate through it via a dedicated MLP, or through spherical basis computation.

In the KiloNeRF scenario, the maximum resolution of each facet in the scene is already calculated, and no greater resolution will become available if the user decides to 'zoom in'.

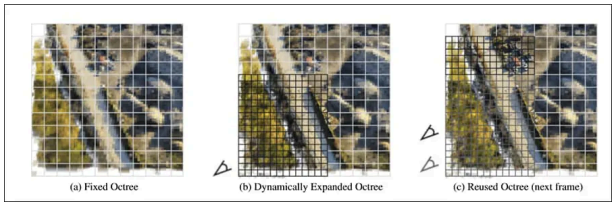
By contrast, NeRF++ can natively handle non-limited, exterior environments by sectioning the potential explorable space into foreground and background regions, each of which is overseen by a dedicated MLP model, which performs ray-casting prior to final composition.

Finally, NeRF in the Wild, which does not directly address unlimited spaces, nonetheless improves image quality in the Phototourism dataset, and its appearance embeddings have been followed in the architecture for Mega-NeRF.

The authors concede also that Mega-NeRF is inspired by Structure-from-Motion (SfM) projects, notably Washington University's Building Rome in a Day project.

Temporal Coherence

Like PlenOctree, Mega-NeRF precomputes a rough cache of color and opacity in the region of current user focus. However, instead of computing paths each time that are in the vicinity of the calculated path, as PlenOctree does, Mega-NeRF 'saves' and reuses this information by subdividing the calculated tree, following a growing trend to disentangle NeRF's tightly-bound processing etiquette.

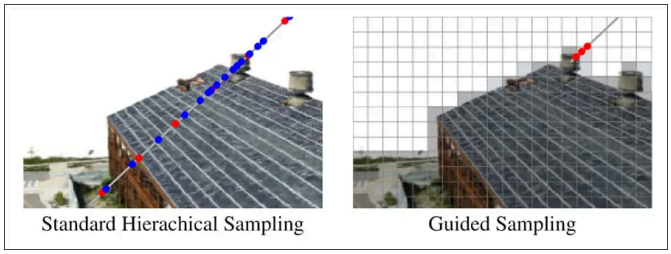


On the left, PlenOctree's single-use calculation. Middle, Mega-NeRF's dynamic expansion of the octree, relative to the current position of the fly-through. Right, the octree is reused for subsequent navigation.

This economy of calculation, according to the authors, notably reduces the processing burden by using on-the-fly calculations as a local cache, rather than estimating and caching them all pre-emptively, according to recent practice.

Guided Sampling

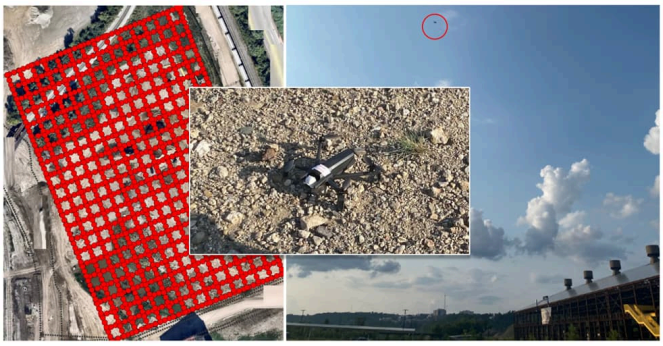
After initial sampling, in accord with standard models to date, Mega-NeRF enacts a second round of guided ray-sampling after octree refinement, in order to improve image quality. For this, Mega-NeRF uses only a single pass based on the existing weights in the octree data structure.



As can be seen in the image above, from the new paper, standard sampling wastes calculation resources by evaluating an excessive amount of the target area whereas Mega-NeRF limits the calculations based on a knowledge of where geometry is present, throttling calculations above a pre-set threshold.

Data and Training

The researchers tested Mega-NeRF on various datasets, including the two aforementioned, hand-crafted sets taken from drone footage over industrial ground. The first dataset, *Mill 19 – Building*, features footage taken across an area of 500 x 250 square meters. The second, *Mill 19 – Rubble*, represents similar footage taken over an adjacent construction site, in which the researchers placed dummies representing potential survivors in a search-and-rescue scenario.



From the paper's supplemental material: Left, the quadrants to be covered by the Parrot Anafi drone (pictured center, and in the distance in the right-hand photo).

Additionally, the architecture was tested against several scenes from UrbanScene3D, from the Visual Computing Research Center at Shenzhen University in China, which consists of HD drone-captured footage of large urban environments; and the Quad 6k dataset, from Indiana University's IU Computer Vision Lab.

Training took place over 8 submodules, each with 8 layers of 256 hidden units, and a subsequent 128 channel ReLU layer. Unlike NeRF, the same MLP was used to query coarse and refined samples, lowering the overall model size and permitting the reuse of coarse network outputs at the subsequent rendering stage. The authors estimate that this saves 25% of model queries for each ray.

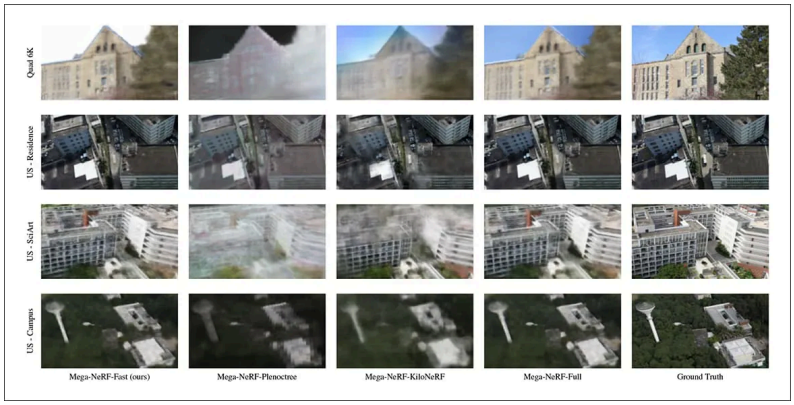
1024 rays were sampled per batch under Adam at a starting learn rate of 5×10^{4}, decaying to 5×10^{-5}. The appearance embeddings were handled in the same way as the aforementioned *NeRF in the Wild*. Mixed precision sampling (training at lower precision than 32-bit floating point) was used, and the MLP width fixed at 2048 hidden units.

Testing and Results

In the researchers' tests, Mega-NeRF was able to robustly outperform NeRF, NeRF++ and DeepView after training for 500,000 iterations across the aforementioned datasets. Since the Mega-NeRF target scenario is time-constrained, the researchers allowed the slower prior frameworks extra time beyond the 24-hour limit, and report that Mega-NeRF still outperformed them, even given these advantages.

	Mill 19 - Building				Mill 19 - Rubble				Quad 6k			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	18.21	0.406	0.665	59:51	20.30	0.433	0.656	60:21	16.29	0.395	0.704	62:48
NeRF++	18.70	0.444	0.596	89:02	20.65	0.474	0.588	90:42	16.43	0.396	0.698	90:34
DeepView	10.52	0.184	0.645	44:45	10.62	0.238	0.684	44:51	9.951	0.283	0.695	39:51
Mega-NeRF	**19.67**	**0.482**	**0.567**	**29:49**	**22.65**	**0.494**	**0.584**	**30:48**	**17.69**	**0.527**	**0.660**	**39:43**
	UrbanScene3D - Residence				UrbanScene3D - Sci-Art				UrbanScene3D - Campus			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	17.77	0.434	0.658	62:40	16.71	0.441	0.723	90:48	17.60	0.392	0.741	61:56
NeRF++	18.04	0.474	0.614	90:48	16.91	0.483	0.664	95:00	17.49	0.393	0.744	93:50
DeepView	11.50	0.119	0.674	51:46	10.28	0.289	0.784	47:54	13.26	0.237	**0.680**	52:48
Mega-NeRF	**20.68**	**0.535**	**0.580**	**27:20**	**19.12**	**0.550**	**0.622**	**27:39**	**18.75**	**0.429**	0.770	**29:03**

The metrics used were Peak signal-to-noise ratio (PSNR), the VGG version of LPIPS, and SSIM. Training took place on a single machine equipped with eight V100 GPUs – effectively, on 256GB of VRAM, and 5120 Tensor cores.



Sample results from the Mega-NeRF experiments (please see the paper for more extended results across all frameworks and datasets) show that PlenOctree cau while KiloNeRF produces artifacts and generally more blurry results.

The project page is at https://meganerf.cmusatyalab.org/, and the released code is at https://github.com/cmusatyalab/mega-nerf.

First published 21st December 2021.

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More