Journées Françaises de l'Informatique Graphique 2023 / R.Vergne and N.Mellado.
(Paper Chairs)

# CompoNeRF: Visualizing Scenes with Objects of Interest by Composing Neural Radiance Fields

Quoc-Anh Bui(1,2), Camille Migozzi(1), Gilles Rougeron(1), Géraldine Morin(2), Simone Gasparini(2), Xavier Aubert (3)

[1]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
[2]Université de Toulouse, Toulouse INP – IRIT, France
[3]Lay3rs SAS, Paris, France

**Abstract**

*This paper introduces a new approach using Neural Radiance Field to explore large scenes containing objects of interest. The input views are partitioned into two groups: scene and object. The first group represents the general scene with one or more NeRFs, while the second one uses a single NeRF per object of interest for more accurate representations,* e.g.*, in the context of cultural heritage preservation. The generation of novel views is achieved by inferring both groups and selecting one of the inferred colors per pixel based on the estimated depth. The method has been tested on both synthetic and real-world datasets.*

**Keywords:** 3D digital twin, cultural heritage, Neural Radiance Field, level of detail

**CCS Concepts**
**•Computing methodologies** → **Image-based rendering;** *Learning latent representations;* **Shape representations; Appearance and texture representations; Virtual reality; Interactive simulation; •Hardware** → *Displays and imagers;*

## 1. Introduction

The acquisition of 3D digital twins of cultural heritage for the purposes of archiving, conservation, archaeology or museography has become commonplace. Until now, photogrammetry and Lidar surveying have been the main techniques used. However, a new technique called Neural Radiance Field (*NeRF* [MST*21]) has recently emerged. It is a novel view synthesis method based on a neural representation of a scene that is trained from a set of input images with known poses. The apparent simplicity of its principle, coupled with the unprecedented quality of the images it produces, has spurred a tremendous quantity of new contributions and developments.

This paper addresses the challenge of exploring a potentially large-scale area containing objects of interest that a user might want to inspect more closely and for which more, or higher resolution data, may be available. In the cultural heritage domain, examples of such objects could be statues or architectural elements found inside a church or a cathedral. Liu *et al.* [LGL*20] proposed using a NeRF coupled with a sparse voxel grid as an adaptive 3D representation to tightly represent geometrical details. However, using a single NeRF makes it difficult to achieve a high level of detail over large areas without incurring prohibitive computation times or memory requirements. Others [TCY*22, TRS22] suggested partitioning the space and training a NeRF by 3D sub-spaces without considering the presence of objects of interest. Our method relies on separating NeRFs into two groups: a scene group composed of a single or multiple NeRFs representing the entire scene with low to medium level of detail, and an object group with a NeRF per object that provides improved, local resolution for close-ups. These groups use different sets of images created during acquisition, with overall views for the scene group and more focused views of objects of interest for the object group.

In the rest of this paper, we will provide a brief overview of the NeRF principle in Section 2; then introduce the concepts and implementations we will be using, and explain how to train the two NeRF groups and combine their inference to create a new view in Section 3. Subsequently, we will present our results on both synthetic and real-world datasets in Section 4. Next, the limitations of the method and perspectives for improvements are described in Section 5. Finally, Section 6 will conclude the paper.

## 2. Neural Radiance Fields

The NeRF method takes as input a set of images with known poses (*i.e.*, position, orientation, field of view angle, *etc.*). It relies on the generation of images by differentiable volume rendering, in which rays emitted from cameras traverse a finite cubic volume. Pixel colors can be obtained by accumulating densities $\sigma$ and colors $\mathbf{c} = (r, g, b)$ inferred at samples along the rays using a Multi-Layer Perceptron (MLP) network that inputs the 3D position $\mathbf{x} = (x, y, z)$ of the sample and the direction $\mathbf{d} = (\theta, \phi)$ of the ray. During training, the squared errors measured from pixels in the known input

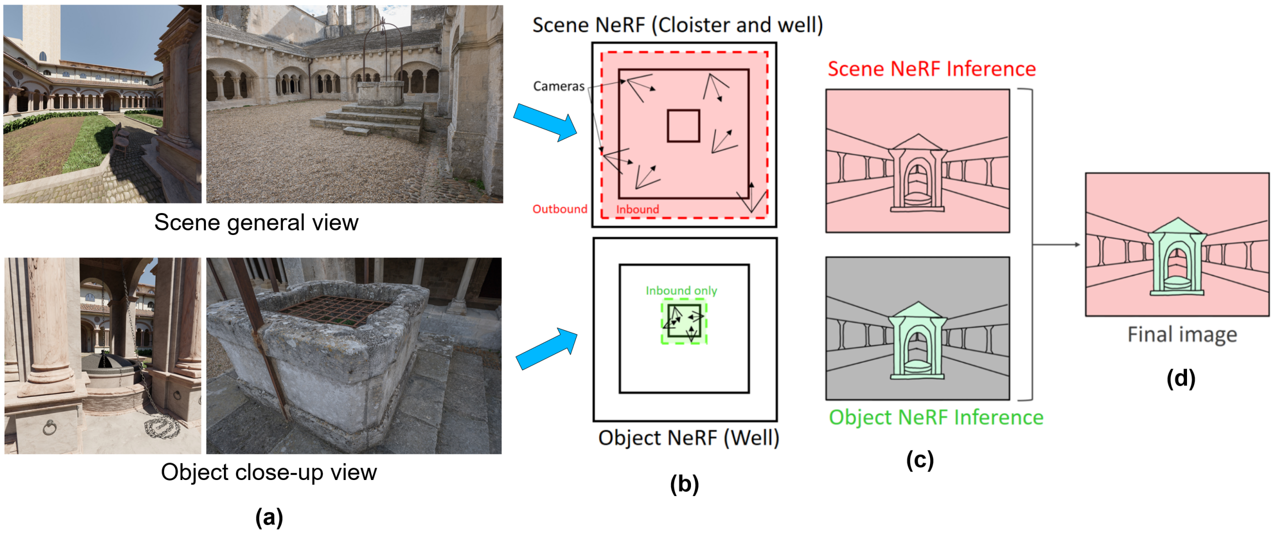



Figure 1: **CompoNeRF Principle. (a)** View types. **(b)** Scene and object NeRF training on two subgroups of images and camera poses with their respective AABBs. **(c)** Rendered images from Scene and object NeRFs. **(d)** Composite image achieved through multi-inference.

images are back-propagated through differentiable volume rendering to update the MLP weights by gradient-retro propagation. As a result, NeRF MLP learns a latent representation of the geometry and directional appearance of the scene. At inference, novel views from unknown new camera poses can be generated using the same direct volume rendering technique.

## 3. Proposed method

### 3.1. Platform

Our implementation is based on Nerfstudio [TWN*23], a Python software platform that embeds different NeRF flavors, including Nerfacto. This one implements a state-of-the-art NeRF method that incorporates the most interesting improvements suggested by the literature and is also the default recommended method on Nerfstudio due to its superior performance. In particular, by implementing ideas from [BMV*22], it is capable of handling unbounded scenes where cameras can view in any direction inside a volume and observe a background beyond the volume. Additionally, it uses a 3D multi-level uniform grid structure of features inspired by [MESK22]. This structure enables faster computation times during both the training and inference stages by significantly reducing the MLP size. It also allows the level of detail representation of the scene to be controlled by adjusting two parameters: $N_{max}$, the finest resolution per axis of the highest grid level, and $T$, the fixed size of the hash tables on the GPU containing the features per level.

### 3.2. Learning overview

In our setting, we consider general views of the scene and close up on objects of interest like shown in Figure 1a. Our goal is to use both image types during training to target different NeRFs/MLPs, and adequately merge the information from the different sources/MLPs during inference. During acquisition, for ease of use, general views are typically taken with cameras equipped with wide-angle lenses or 360 panoramic cameras. These views are used to train a set of scene NeRFs. In this paper, only one NeRF with unbounded properties is used. The finite volume of the NeRF is defined as the Axis Aligned Bounding Box (AABB) enclosing all camera positions and extended by a small margin (shown as a dotted red line in Figure 1b). Moreover, close-up views of objects of interest are taken with cameras potentially equipped with lenses with lower near-field distance and a narrower field of view. These images are used to train a bounded NERF whose volume is also the AABB enclosing the object camera poses (shown as a dotted green line in Figure 1b). In this preliminary work, only one object is considered. Note that the object's geometry and appearance are thus learned by both scene NeRF and object NeRF.

### 3.3. Multi-Inference

In order to generate a novel view, both scene and object NeRFs must infer an image each, as shown in Figure 1c. When creating the final image, a color is selected for each pixel based on its estimated depth. If the first point on the surface along the camera's ray is within the object NeRF's AABB, then the pixel color is inferred from that NeRF. Otherwise, the scene NeRF pixel color is used. This method is effective for handling objects with intricate geometry, concave shapes, or even holes, as shown in Figure 2.

## 4. Results

### 4.1. Datasets

**Lone Monk.** A synthetic dataset of images was generated using Blender from the *SILVR* dataset [CAP*22] for the Lone Monk scene. The scene NeRF was trained from 10400 images taken throughout the entire cloister and looking all around, while the object NeRF was trained from about 700 images rendered with

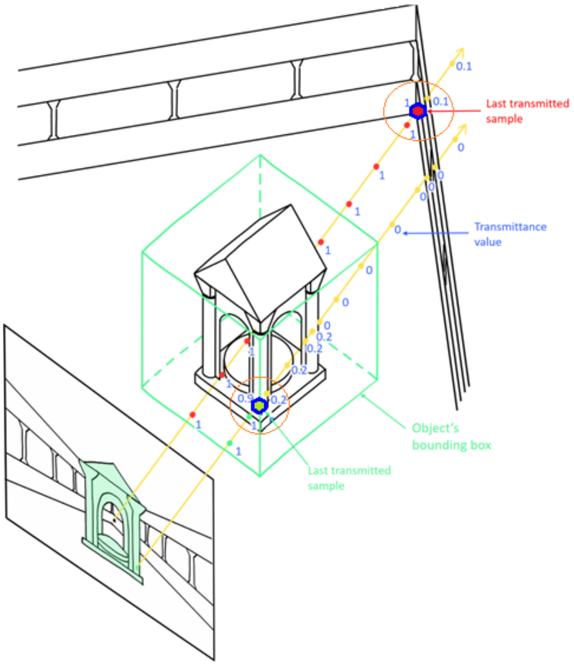



Figure 2: **Multi-Inference**. The green first contact point is within the object NeRF's AABB, the pixel color is inferred from that NeRF. Otherwise, red point, the scene NeRF pixel color is used.

the same camera parameters, focusing on the well at the center. All images are rendered at $2000 \times 2000$ pixels from viewpoints sampled on a path that begins around the well and then goes around the cloister and finally into the corridors and ends at the starting point.

**Montmajour.** Our method was tested on a very similar real-world scene, a cloister in Montmajour, a medieval abbey located in the south of France. We captured 920 images with 20 and 24 mm wide-angle lenses, which were then used to train the scene NeRF. And 180 images of the well were taken with 24 and 50 mm lenses and used to train the object NeRF. All images are at $6048 \times 4024$ pixels resolution.

On both datasets, we choose 15 overall images containing the object of interest for testing. The poses of all the cameras were retrieved in a single reference system using a standard Structure from Motion algorithm from Metashape.

### 4.2. Comparisons

To evaluate our model, we compare it against Nerfstudio's default state-of-the-art method, Nerfacto. The baseline method chosen is Nerfacto-huge, a larger parameter version of Nerfacto. It uses $N_{max} = 16384$ and a hash table size of $T = 23$. We train this model for $100K$ iterations with all the images, including scene general and object close-up views.

Our CompoNeRF method consists of scene NeRF and object NeRF, both based on the Nerfacto model and both utilizing the same hash table size ($T = 23$). However, the object NeRF employs a higher definition with $N_{max} = 16384$, while the scene NeRF uses $N_{max} = 8192$. The smaller AABB on the object NeRF implies a higher level of detail for the object of interest. We train these two models separately, each for $50K$ iterations, which is half the number of iterations used in the baseline. We opted for the highest definition and twice the number of iterations as the baseline to ensure an equal and fair comparison with our CompoNeRF models.

We first evaluate our method on the synthetic Lone Monk dataset. As shown in Figure 3a, CompoNeRF is capable of rendering small details on the well, such as the chain in the middle. However, our model still has limitations, as it misses the end of the chain which disappears into the well while it is not the case for the ground truth. Furthermore, we observe a higher level of detailed texture on the pillar when compared to the baseline method.

Next, the methods are evaluated on the real-world Montmajour dataset. With CompoNeRF, the well and the ground in its surroundings are rendered very precisely, while the rest of the cloister, including the farther background, is displayed with a coarser level of detail see Figure 3b. In particuliar, we can observe more detail on the surface of the well and on the gravel particles on the stone shelf than in Nerfacto-huge.

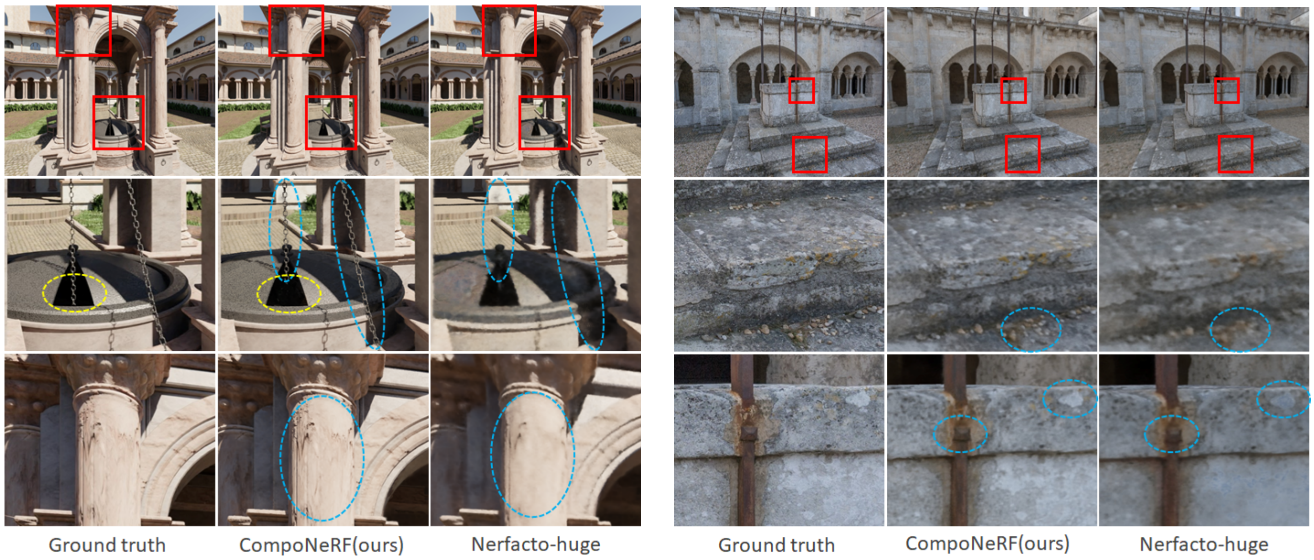
| **Method** | Training time (h) | Rendering time (s) | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|
| CompoNeRF | 4.5 | 12.8 | **26.79** | **0.820** | **0.214** |
| Scene NeRF | **3.3** | 6.5 | 25.52 | 0.791 | 0.267 |
| Nerfacto-huge | 6.6 | 6.5 | 25.69 | 0.790 | 0.246 |

Table 1: A **quantitative comparison** of methods on the *Lone Monk* scene

| **Method** | Training time (h) | Rendering time (s) | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|
| CompoNeRF | 9.9 | 103.7 | **19.80** | **0.523** | **0.566** |
| Scene NeRF | **5.6** | **52.3** | 19.02 | 0.500 | 0.695 |
| Nerfacto-huge | 11.5 | 53 | 19.41 | 0.510 | 0.648 |

Table 2: A **quantitative comparison** of methods on the *Montmajour* scene

For each dataset, Table 1 and Table 2 compare the methods, showing the rendering quality (PSNR, SSIM, and LPIPS metrics) as well as the training and rendering time. We can see that our CompoNeRF outperforms the baseline approach when averaged across scenes. In our setup, we train all methods using a single 80 GB A100 GPU on a DGX station. The total training time for both the scene and object NeRFs of our method is faster than the entire training time of the baseline method (10–20% faster). However, it's important to note that the need to render the same image twice makes the image rendering time double compared to the baseline method.




(a) Test-set views of synthetic **Lone Monk** scene

(b) Test-set views of real-world **Montmajour** scene

Figure 3: We compare the scene **rendering quality** with the Nerfacto-huge baseline on both datasets.

## 5. Limitations and Perspectives

Despite its superiority, our model also has limitations.

First, images focused on the object also capture the surrounding area; however, object NeRF may not accurately infer this area as the camera poses were not specifically focused on it. As a result, when composing the final image, blurry and poorly rendered boundaries may occur around the object. A possible solution involves tightening the AABBs around the objects. To achieve this, we could perform segmentation on 2D images to reconstruct consistent masks for objects of interest, inspired by *Object-NeRF* [YZX*21]. With these object masks, AABBs could be refined either by intersecting 3D mask frustum photogrammetry or by propagating image mask IDs to 3D sparse point cloud extracted from SfM algorithms, by taking inspiration from *K3BO* [JRZ23]. Another approach is to employ 3D segmentation after estimating the object's geometry through rapid training of a NeRF model on the object.

In the scenario with multiple objects of interest in the scene, rendering time will be proportional to the number of objects, resulting in a significant increase in the total time required for final image composition. A straightforward optimization would be to discard objects that are not within the current view frustum. This could be easily and efficiently performed using the tightened AABBs mentioned above. Similarly, the resolution of distant objects could be inferred by the scene NeRF model, as long as a lower or medium level of detail image is sufficient.

In order to handle large-scale scenes, we could generalize the use of multiple NeRF models with space partitioning techniques inspired by *e.g.*, *Block-NeRF* [TCY*22], *Mega-NeRF* [TRS22].

In the real-world Montmajour dataset, alongside the RGB image data, a 3D point cloud scan from LiDAR with millimeter precision was also captured. For the primary focus of this paper, we exclusively utilize the RGB image data to maintain consistency for comparison with current state-of-the-art methods. Nevertheless, the opportunity exists to leverage the geometric information provided by the 3D point cloud in training NeRFs, like in *DS-NeRF* [DLZR22] or *PointNeRF* [XXP*22], to potentially achieve improved performance.

Finally, up to this point, the division of images into groups has remained a manual process, resulting in still having images focused on the object of interest within the scene group. Subsequently, a simple algorithm will be developed to identify all cameras that view the object of interest AABB in sufficient detail, grouping them together for the training of the corresponding object NeRF.

## 6. Conclusion

A new technique has been introduced for virtually exploring scenes offering improved resolution for objects of interest, for which detailed input images are provided. CompoNeRF involves multiple NeRFs that generate images for general views as well as detailed views of specific objects. This approach takes advantage of the fast training capabilities of recent state-of-the-art methods and successfully achieves high rendering quality. Moreover, by modifying an existing NeRF implementation, such as the one suggested by *Instant-NGP* [MESK22], or even adapting a non NeRF method such as *3D Gaussian Splatting* [KKLD23], our method provides the framework for interactive visits to complex heritage sites using a VR headset.

**Acknowledgment.**

We would like to thank the Centre des Monuments Nationaux for allowing us to survey the Montmajour Abbey site.

## References

[BMV*22] BARRON J. T., MILDENHALL B., VERBIN D., SRINIVASAN P. P., HEDMAN P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR* (2022). 2

[CAP*22] COURTEAUX M., ARTOIS J., PAUW S. D., LAMBERT P., WALLENDAEL G. V.: Silvr: a synthetic immersive large-volume plenoptic dataset. *Proceedings of the 13th ACM Multimedia Systems Conference* (2022). URL: https://api.semanticscholar.org/CorpusID:248266704. 2

[DLZR22] DENG K., LIU A., ZHU J.-Y., RAMANAN D.: Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022). 4

[JRZ23] JOSHUA M., RAMIN N., ZHANG L.: K3bo: Keypoint-based bounding box optimization for radiance field reconstruction from multi-view images. In *2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (2023), pp. 134–139. doi:10.1109/ICMEW59549.2023.00030. 4

[KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics 42*, 4 (July 2023). URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/. 4

[LGL*20] LIU L., GU J., LIN K. Z., CHUA T.-S., THEOBALT C.: Neural sparse voxel fields. *NeurIPS* (2020). 1

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. 41*, 4 (July 2022), 102:1–102:15. URL: https://doi.org/10.1145/3528223.3530127, doi:10.1145/3528223.3530127. 2, 4

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 1

[TCY*22] TANCIK M., CASSER V., YAN X., PRADHAN S., MILDENHALL B., SRINIVASAN P., BARRON J. T., KRETZSCHMAR H.: Block-NeRF: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). 1, 4

[TRS22] TURKI H., RAMANAN D., SATYANARAYANAN M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 12922–12931. 1, 4

[TWN*23] TANCIK M., WEBER E., NG E., LI R., YI B., KERR J., WANG T., KRISTOFFERSEN A., AUSTIN J., SALAHI K., AHUJA A., MCALLISTER D., KANAZAWA A.: Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings* (2023), SIGGRAPH '23. 2

[XXP*22] XU Q., XU Z., PHILIP J., BI S., SHU Z., SUNKAVALLI K., NEUMANN U.: Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5438–5448. 4

[YZX*21] YANG B., ZHANG Y., XU Y., LI Y., ZHOU H., BAO H., ZHANG G., CUI Z.: Learning object-compositional neural radiance field for editable scene rendering. In *International Conference on Computer Vision (ICCV)* (October 2021). 4