Custom Background for NeRF

Junhua Ma
UC Berkeley
junhua@berkeley.edu

Abstract

Neural Radiance field (NeRF) models allow synthesis of new views of an object from different view directions, and many prior works have focused on enhancing the object in the rendered views but not the background. Utilizing diffusion models, we present a simple and efficient pipeline that allows synthesis of views of object with custom background that is consistent across different view directions.

1. Introduction

Neural radiance field (NeRF) introduced by Mildenhall et al. [3] is a popular method used to generate views from novel view directions of an object. Given a set of training data consisting of different camera positions, orientations, and views as images, the NeRF model learns to predict the RGB color and density at every point of a ray given a ray center and direction, which can then be converted into pixel RGB values through a volumetric rendering function. While there are simple methods for injecting custom background to the synthesized views from NeRF models, this results in fixed backgrounds with only the objects appearing in different orientations. And for full-fledged interfaces like NeRF Studio with Blender plugins for adding custom backgrounds and directly placing NeRF model into a 3D scene [4], these tools would require work to setup and may not support diffusion-based 3D scene synthesis to automatically synthesize 3D backgrounds.

In this paper, we present a simple to use and efficient pipeline that only requires a text prompt as input, and can then generate views of the object with backgrounds that suit the prompt while being consistent between different view directions. In other words, the object should appear to be placed into a generated 3D scene, and the background would change with the object based on the view direction. With this pipeline, assuming the users have access to the NeRF model of an object, they can quickly generate novel views of the object from any view direction with a desired background without needing to resort to more advanced tools or do any manual work.

2. Methodology

The proposed pipeline consists of two major stages as shown in Fig. 1. In the first stage, the input text prompt is passed into the Flux-360 [2] spherical panorama diffusion model to generate spherical panoramas that fit the prompt. In the second stage, views are extracted from the generated spherical panorama based on the desired view direction, and injected as background image in the volumetric rendering step of NeRF. The details of all major components of the pipeline after obtaining the spherical panorama are covered step by step in the following subsections.

2.1. Generating Views from Spherical Panorama

Typically, to render 2D views from a spherical panorama, the center of projection is set at the center of sphere, and images are obtained by projecting rays to hit points on the sphere which are used as pixel values for the corresponding pixels. However, to generate a background image for NeRF synthesized views, having the center of projection at the center of sphere would lead to inconsistencies in the object's position in relation to the scene when the object and the background rotate together. Since the NeRF model is trained with the object being at the center of sphere, the ground that the object should stand on is completely outside of views generated with projection from center. Therefore, an initial setup is to have the center of projection be a point on the sphere instead to render views as shown in Fig. 2. This makes it possible to have the foreground where the object stands on as well as the background in view simultaneously to help generate better and more consistent backgrounds.

Specifically, the spherical panorama view rendering setup is implemented as follows. Given a position on the unit sphere denoted by $x_0 = (\theta, \phi)$ as the center of projection, the direction of projection or view direction d that is orthogonal to the image plane, and d_u orthogonal to d that always points downward relative to the image plane, a 2D grid of (u, v) coordinates is formed as the image plane at a fixed distance away from the x_0 in the view direction, where a ray can then be formed between x_0 and every point (u, v) on the image plane, with the direction d_{uv} of each ray

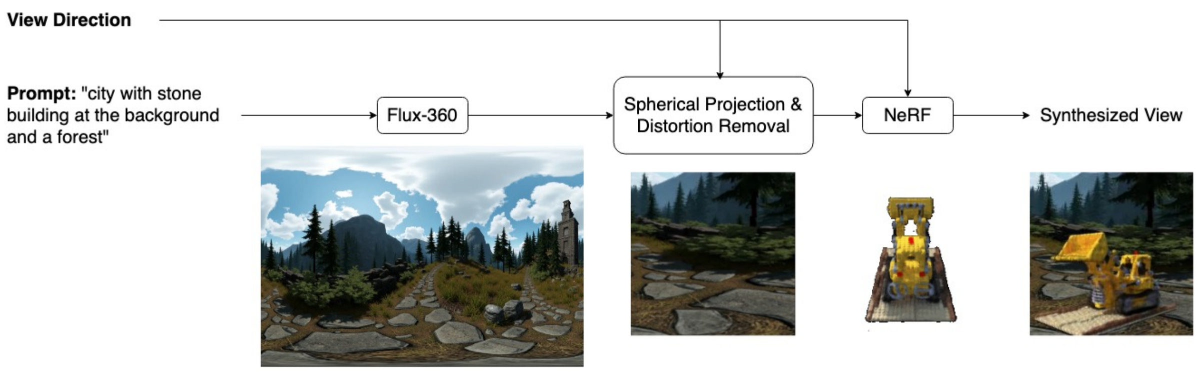
View Direction
Prompt: "city with stone building at the background and a forest"
Flux-360
Spherical Projection & Distortion Removal
NeRF
Synthesized View

Figure 1. Overview of pipeline.

computed as:

$$d_{uv} = d + d_u \cdot u + d_v \cdot v \tag{1}$$

where $d_v = d \times d_u$. Afterwards, each pixel (u, v) in the image plane obtains its RGB value by projecting the corresponding ray specified by the function $x_{uv}(t) = x_0 + d_{uv}t$ onto the sphere. This process amounts to solving for t such that $||x_{uv}(t)|| = 1$ for the unit sphere, which can be simplified down to the following quadratic equation

$$||d_{uv}||^2 t^2 + 2x_0 d_{uv} t + ||x_0||^2 - 1 = 0 \tag{2}$$

which can be efficiently solved to obtain t and the projected point on sphere $x_{uv}(t)$, which is then converted from spherical to cartesian coordinates (u_p, v_p). Bilinear interpolation is applied to round to integer values and the pixel value is obtained from the spherical panorama image as $im_{panorama}(u_p, v_p)$.

2.2. Distortion Removal

Since the center of projection is no longer at the center, one issue with the rendered views is distortion, which is especially worse for certain view directions. For example, for center of projection close to ground level ($\phi = 90°$) with horizontal view direction, the rendered views would have a stretched foreground and U-shaped distortion for the more distant background.

A simple but surprisingly effective approach to remove such distortions is depicted by Fig. 3. Instead of directly projecting rays and rendering image with the final desired shape, we first project onto an image plane with smaller width and greater height. The center of projection and view direction is also modified to account for the greater vertical view. We then compress the image down to the final desired shape using the Lanczos method to obtain the final rendered view which has much less distortion. Some samples comparing views with and without distortion removal are shown in Fig. 4.

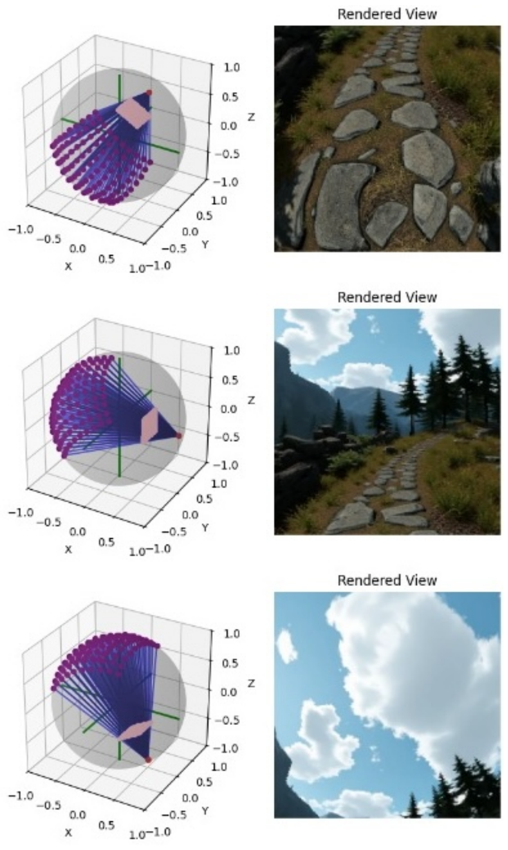

Figure 2. Illustration of projection view rendering setup and results, view directions are all from respective center of projection towards center of sphere. The red dot is the center of projection, pink is the image plane, and purple dots are projected points on sphere. The rays are down-sampled for display. **Top:** COP ($\theta = 90°, \phi = 30°$). **Middle:** COP ($\theta = 90°, \phi = 90°$). **Bottom:** COP ($\theta = 90°, \phi = 150°$).

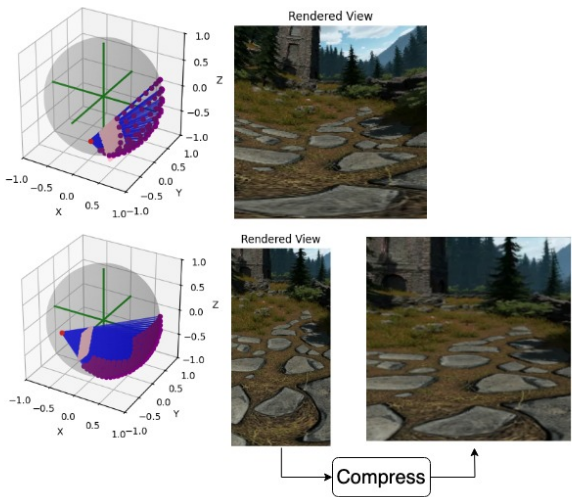

Figure 3. Overview of the view rendering with or without distortion removal. **Top:** View rendering without distortion removal. **Bottom:** View rendering with distortion removal.

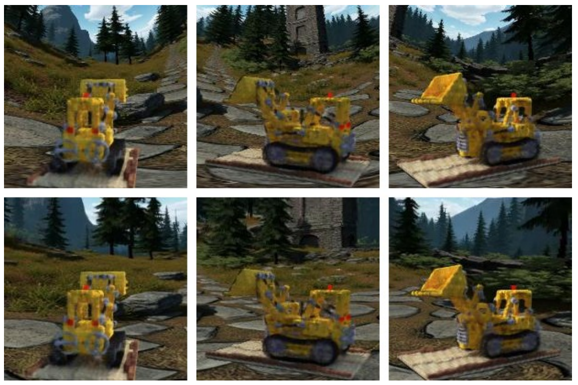

Figure 4. Comparison between rendered views with and without distortion removal. **Top:** Without distortion removal. **Bottom:** With distortion removal.

2.3. NeRF Background Injection

With the background image im_{bg} rendered from spherical panorama, we modify the volumetric rendering formula of NeRF to inject it into the synthesized views. The NeRF volumetric rendering equation [3] converts the model outputs of predicted RGB values and density of points along a ray into the RGB value of the pixel corresponding to the ray in the synthesized view. Mathematically, the rendered pixel color C along a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is defined as:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt \tag{3}$$

where $T(t)$ is the accumulated transmittance:

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right) \tag{4}$$

The modified volumetric rendering equation with background image injection is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt + T(t_f)im_{bg}(\mathbf{r}) \tag{5}$$

where $im_{bg}(r)$ is the background image pixel value corresponding to ray $\mathbf{r}$.

2.4. Other Adjustments

Since NeRF models generally renders views with the object at the center of the image regardless of view direction, some adjustments of the position of the NeRF object is needed to make it fit together with the generated background. This is especially true when the center of projection close to the ground like $\phi = 80°$ where the ground is close to the bottom of the image, and without any adjustments the NeRF object would appear to be floating above ground. The implementation of this pipeline uses a simple method by shifting the rendered background image up by a certain amount before passing to NeRF volumetric rendering, and then shifting the final output image with the NeRF object back down. However, an alternative and potentially better approach may be to instead shift the center of ray projection for NeRF upward to directly synthesize views of object that appear closer to bottom of image.

3. Results

A sampled set of results of the full pipeline is shown in Fig. 5, with different input prompts to generate different backgrounds using the same NeRF model and a selected set of view directions with fixed center of projection at $\phi = 80°$ for the best views. Additionally, we show a set of results that doesn't look as good due to distortions or inaccurate panorama generation in Fig. 6, the details will be covered with other limitations of the pipeline in the discussion.

We also present synthesized views at various view directions in Fig. 7 with different settings of ϕ to cover the entire hemisphere of possible view directions.

4. Discussion

Overall, the results of the pipeline are good enough to meet the goal of having consistent background at various view directions, with the object appearing to be placed at a fixed location in a scene depicted by the generated spherical panorama. One notable highlight of the outputs is how the object overall does appear to be placed on the ground instead of hovering in air, and works well with the distortion removal setup that produces a visually coherent flat ground plane in the background image for the object to stand on. Additionally, the 360 degree view of the spherical panorama is utilized fully to generate views at all possible view directions which complements NeRF.

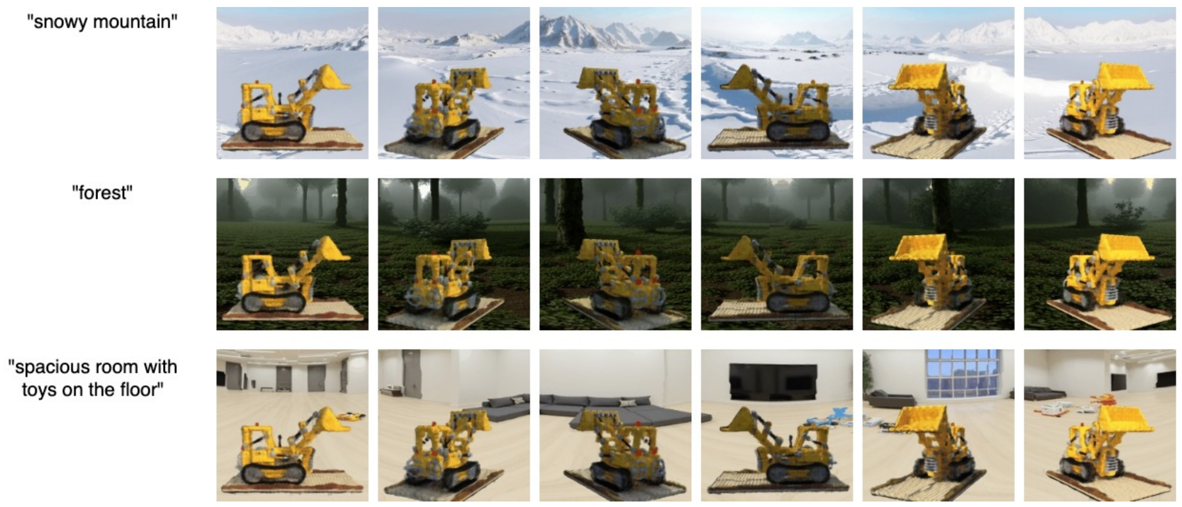

Figure 5. Results of pipeline at selected view directions given different input prompts.

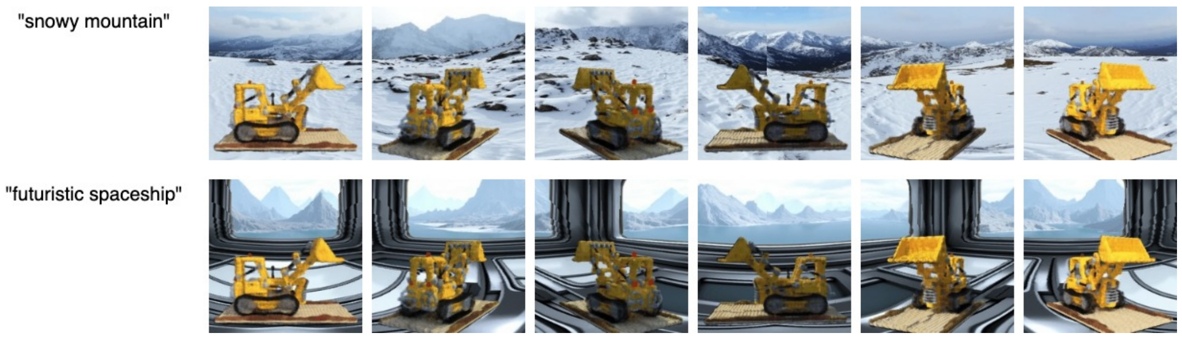

Figure 6. Failed results of pipeline due to issues with generated panorama and distortion.

However, there are some important limitations with the method and pipeline. As shown by the failed results, any small issues with the generated spherical panorama like misalignment or flawed geometry with the scene would directly lead to poor results. Additionally, distortion still becomes quite notable if the scene depicts mostly straight lines and very close to the NeRF object, as the distortion removal work best for more distant background distortions and is also not enough to fully remove distortions. Finally, for synthesizing views at various view directions, changing θ value of the center of projection simply rotates the camera horizontally about the object which generally doesn't require any additional setup. For fixed ϕ, if the view looks good from one θ position, then the views should be good for any other θ positions. However, due to the large amount of possible parameters in the pipeline, we have not found a more generalized setup that automatically works for all ϕ angles for the center of projection. Therefore, some manually tuning of parameters is needed to generate good views at different ϕ, and some slight inaccuracies with parameter settings may result in slight inconsistencies with the object's position and orientation in relation to the scene. Since there are some notable patterns in the parameter change that correspond to different ϕ values, this suggests the possibility of generalization.

For future directions, one major aspect that is missing from this work overall is the integration of lighting and shadows. The NeRF object may have a different lighting and dynamic range than the generated panorama which makes the rendered views less realistic and appealing. One potential method to explore would be to identify light source from the generated panorama as a 3D point, utilize methods like intrinsic decomposition [1] to obtain albedo and reflective properties of the NeRF object, and perform relighting on the NeRF object to fit the lighting of the scene. The shadows may similarly be obtained through simulating ray projection from the light source.

5. Conclusion

In this paper we presented a simple and efficient pipeline to inject custom background to NeRF synthesized views that is consistent between view directions. We utilized ad-

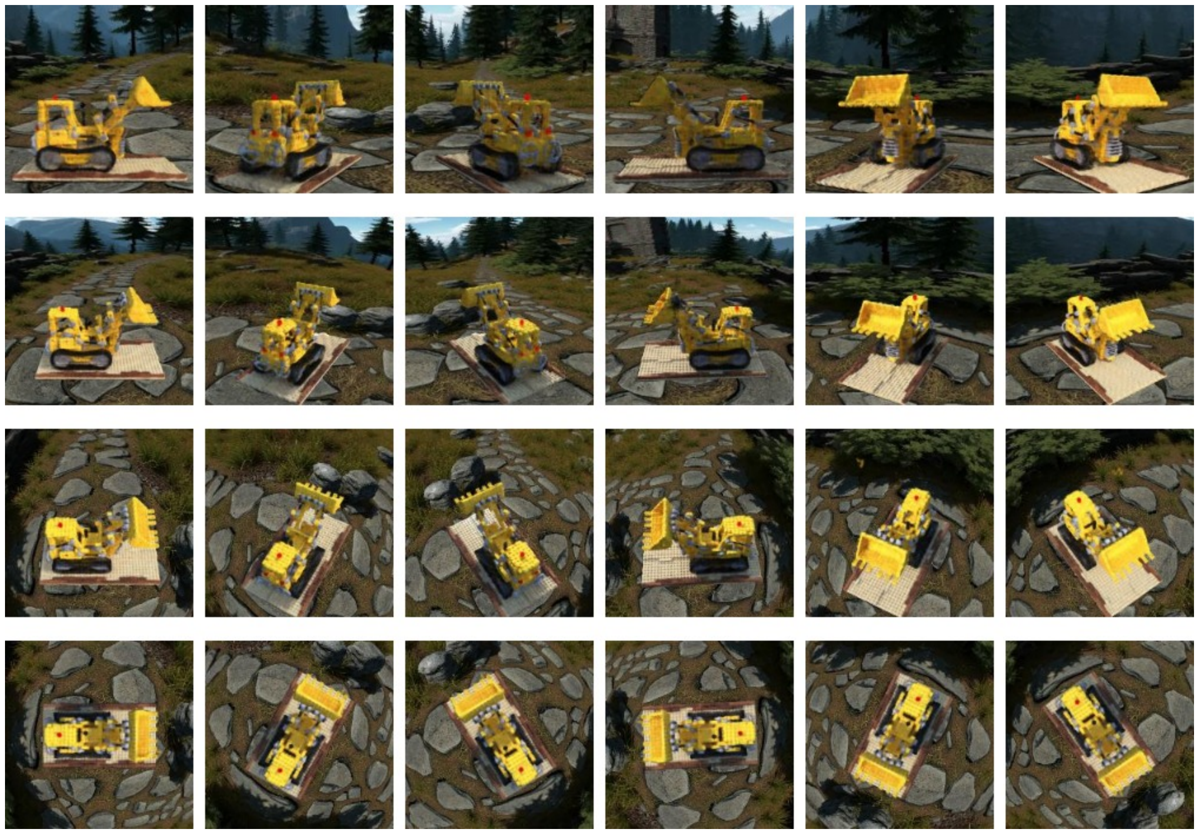

Figure 7. Results of pipeline at selected view directions, each row corresponds to a different ϕ for center of projection with $\phi = 80^\circ$, 60°, 30°, 0° from top row to bottom row respectively.

vancements in generative models to generate custom backgrounds from text prompt inputs, and obtained decent views from all view directions through an effective projection setup to render 2D views from spherical panoramas and remove distortions. Lighting and shadows are not accounted for in the presented pipeline which would further improve the results as a future direction.

References

[1] Chris Careaga and Yağız Aksoy. Intrinsic image decomposition via ordinal shading. *ACM Trans. Graph.*, 43(1), 2023. 4

[2] Ignacio Gorriti. Flux-360. https://replicate.com/igorriti/flux-360, 2024. Accessed: 2024-12-15. 1

[3] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1, 3

[4] NeRF Studio Team. Nerf studio documentation. https://docs.nerf.studio/, 2024. Accessed: 2024-12-03. 1