Neural Radiance Fields Convert 2D to 3D Texture

Zheng Lin[1], Chenghao Wang[2], Zichao Li[3], Zhuoyue Wang[4], Xinqi Liu[5] and Yue Zhu[6]
[1]University of California, Santa Cruz
[2]Georgia Institute of Technology, United Sates
[3]Canoakbit Alliance Inc., Oakville, Canada
[4]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United Sates
[5]Carnegie Mellon University, United Sates
[6]Independent, China

[1]Corresponding Author: zeyuwang@ucla.edu

Received: 21-04-2024 Revised: 09-05-2024 Accepted: 29-05-2024

ABSTRACT
The objective of our project is to capture pictures or videos by surrounding a circle of objects, such as chairs, tables, cars, and more.[1]Utilizing advanced 3D reconstruction technology, we aim to generate 3D models of these captured objects. Post reconstruction, these 3D models can be edited through an intuitive interface, enabling users to apply different textures and make other modifications. This project has significant applications in various domains such as home decoration, vehicle customization, and beyond. For the 3D reconstruction in this project, we employed Nvidia's latest ngp-instant method, which leverages hash encoding for 3D graphics reconstruction. This technique offers a faster inference speed compared to traditional NeRF (Neural Radiance Fields). Following the 3D reconstruction, we apply volume rendering to visualize the 3D models. To facilitate user editability[2], we integrated an editable interface inspired by StyleGAN, utilizing a texture loss function to transform the 3D model into a customizable texture. This combination of technologies allows for a seamless and efficient process in creating and editing 3D models from 2D images.

***Keywords**:* texture, 2d images, 3d images, neural radiance

I. INTRODUCTION

In this project, we are exploring the realm of image generation, with a shift in focus towards creating 3D graphics from 2D images. Building upon our understanding of the NeRF model[3], we have incorporated material codes and shape information, which enables us to produce 3D objects through user-driven editing.[4] This approach not only enhances the realism of the generated models but also offers a high degree of flexibility in terms of customization. By capturing images or videos of objects arranged in a circular formation, we are able to gather comprehensive visual data from multiple angles.[5] This data serves as the foundation for our 3D reconstruction process, where we utilize Nvidia's ngp-instant method. This method stands out due to its use of hash encoding, providing a notable improvement in inference speed over traditional NeRF methods.[6] Once the 3D reconstruction is complete, we employ volume rendering to obtain detailed and accurate 3D graphics of the objects. [7]To ensure that these 3D models can be easily edited by users, we referenced the editable interface of StyleGAN[8]. By incorporating a texture loss function, we have enabled the transformation of the 3D model into a modifiable texture. [10]This allows users to apply different textures and make other adjustments, making the process of 3D modeling both accessible and versatile. [11]Our project holds great promise for applications in home and vehicle decoration, among other areas, demonstrating the potential of combining 3D reconstruction with user-friendly editing tools.[12]

II. RELATED WORK

The recent project on 2D reconstruction to 3D mainly focused on the transformation of nerf (Neural Radiance Fields),[13] and we got a lot of inspiration. First of all, the idea of NeRF is to implicitly store the 3D scene in the neural network. We only need to input a camera pose to obtain the scene picture. NeRF models the scene as a continuous 5D radiation field (in fact, it feels like an implicit voxel description).[14]

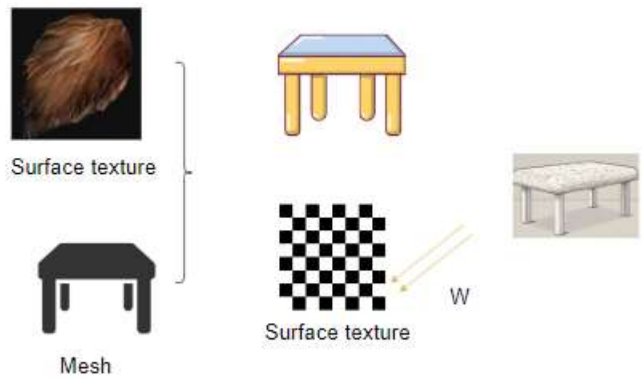


Figure 1: Framework

investigate the use of neural fields for modeling diverse mesoscale structures, such as fur, fabric, and grass.[15] Instead of using classical graphics primitives to model the structure, we propose to employ a versatile volumetric primitive represented by a neural reflectance field (NeRF-Tex)[16], which jointly models the geometry of the material and its response to lighting. The NeRF-Tex primitive can be instantiated over a base mesh to "texture" it with the desired meso and microscale appearance. We condition the reflectance field on user-defined parameters that control the appearance. A single NeRF texture thus captures an entire space.[17]

III. SOLUTION

My purpose is to generate interactive 3D graphics and implement material editing functions. So I use nerf to do 3d reconstruction of 2d graphics.[18] Nerf is to obtain the value of a pixel by accumulating the pixels of light sources from different viewing angles. Volume rendering is then performed to render these values. In this way, the transformation from 2d to 3d process is achieved. Then use UI editing to edit the object, I use GAN rewriting interface to re-edit our object. And we design a material loss function to learn from user edits.[20]

3.1 Methodology

To edit the texture of a instance's part, the user selects a desired texture and scribbles a foreground mask over a rendered view indicating where the texture should be applied. The user may optionally also scribble a background mask where the texture should remain unchanged. These masks do not need to be detailed; instead, a few coarse scribbles for each mask suffice. [21]The user provides these inputs through a user interface, which we discuss in the appendix. Given the desired target texture and foreground/background masks, we seek to update the neural network.

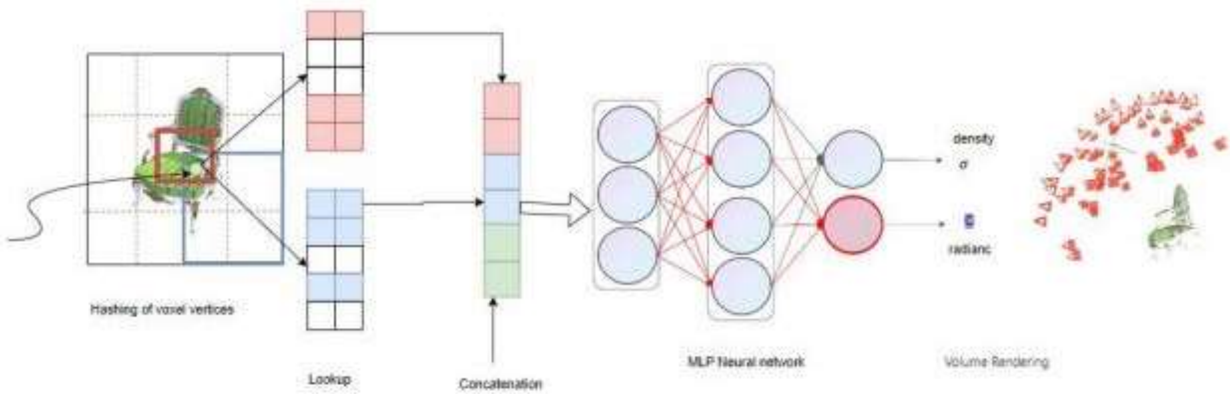
We define cf texture for a ray r at a pixel location within the foreground mask provided by the user scribble and let yf = {(r, cf)} be the set of ray texture pairs provided by the entire user scribble. Furthermore, for a ray r at a pixel location in the background mask, let cb be the original rendered texture at the ray location. let y f = {(r, cb)} be the set of rays and colors provided by the background user scribble

$$\mathcal{L}_{\text{rec}} = \sum_{(\boldsymbol{r},\boldsymbol{c}_f)\in y_f} \|\hat{C}(\boldsymbol{r}, \boldsymbol{z}^{(s)}, \boldsymbol{z}^{(c)}) - \boldsymbol{c}_f\|^2 + \sum_{(\boldsymbol{r},\boldsymbol{c}_b)\in y_b} \|\hat{C}(\boldsymbol{r}, \boldsymbol{z}^{(s)}, \boldsymbol{z}^{(c)}) - \boldsymbol{c}_b\|^2$$

We define our texture editing loss as the sum of our reconstruction loss and our regularization loss

$$\mathcal{L}_{\text{texture}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}$$

Our goal is to allow user edits of a continuous volumetric representation of a 3D scene. In this section, we first describe a new neural network architecture that more accurately captures the shape and appearance of an object class. We then describe how we update network weights to achieve texture and shape editing effects. To achieve this goal, we build upon the recent [22]. While the NeRF representation can render novel views of a particular scene, we seek to enable editing over an entire shape class, e.g., "chairs".



We define x = (x, y, z) be a 3D location, d = (φ, θ) be a viewing direction. The network is parametrized as a multi-layer perceptron (MLP) such that the density output σ is independent of the viewing direction, while the radiance c depends on both position and viewing direction. Then these two parameters are then subjected to volume rendering and we can get a 3D model. And our NGP-intsant method is different from traditional nerf. He has a faster speed because he uses hash coding to make the extracted information more accurate, and uses C++cuda to calculate, which also makes the network faster.[23]

IV. RESULT AND CONCLUSION

We test the material editing function, we edit the RGB of some common materials, such as stone, wood, etc., and then use the RGB value of the pixel and the RGB value drawn by the user to do gradient descent, and set the range constraint.[24]

Figure 4: Add wood material

What I can find is that simply implementing editing materials, our effect is still good. We can now achieve a good implementation of switching objects to wood or stone. However, when we refactor ngp-intsant after editing, the data we output is very scarce due to the limited input data, which will cause our rendering effect to be inferior to professional data sets

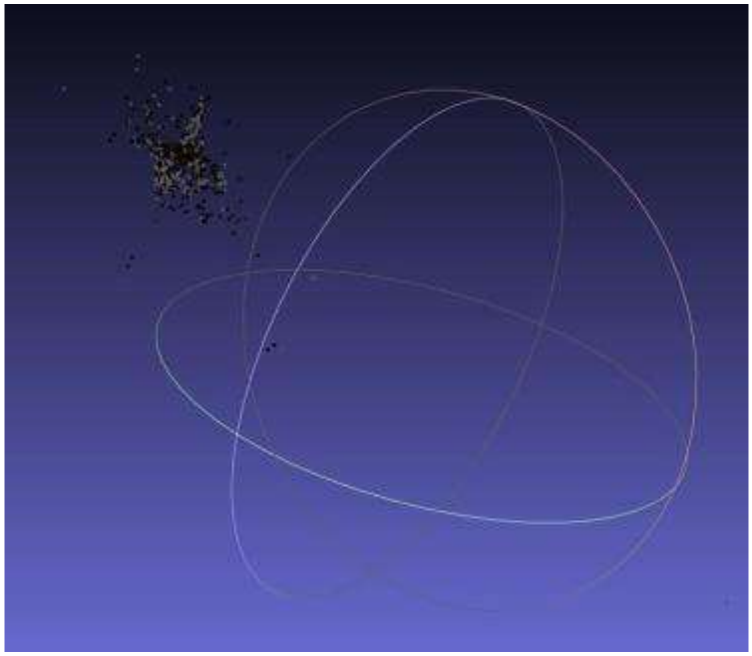


Figure 4: Refactoring after texture editing

REFERENCES

1. Li, Zhenglin, et al. (2023). Stock market analysis and prediction using LSTM: A case study on technology stocks. *Innovations in Applied Engineering and Technology (2023)*, 1-6.
2. Hong, Bo, et al. (2024). The application of artificial intelligence technology in assembly techniques within the industrial sector. *Journal of Artificial Intelligence General Science (JAIGS), 5*(1), 1-12.
3. Zhou, Chang, et al. (2024). Optimizing search advertising strategies: Integrating reinforcement learning with generalized second-price auctions for enhanced ad ranking and bidding. arXiv preprint arXiv:2405.13381.
4. Li, Shaojie, Yuhong Mo, & Zhenglin Li. (2022). Automated pneumonia detection in chest x-ray images using deep learning model. *Innovations in Applied Engineering and Technology*, 1-6.
5. Zhou, Chang, et al. (2024). Optimizing search advertising strategies: Integrating reinforcement learning with generalized second-price auctions for enhanced ad ranking and bidding. arXiv preprint arXiv:2405.13381.
6. Mo, Yuhong, et al. (2024). Password complexity prediction based on roberta algorithm. *Applied Science and Engineering Journal for Advanced Research, 3*(3), 1-5.
7. Jin, Jiajun, et al. (2024). Enhancing federated semi-supervised learning with out-of-distribution filtering amidst class mismatches. *Journal of Computer Technology and Applied Mathematics, 1*(1), 100-108.
8. Dai, Shuying, et al. (2024). AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks. *Journal of Artificial Intelligence General science (JAIGS), 5*(1), 13-21.
9. Mo, Yuhong, et al. (2024). Large Language Model (LLM) AI text generation detection based on transformer deep learning algorithm. *International Journal of Engineering and Management Research, 14*(2), 154-159.
10. Song, Jintong, et al. (2024). A comprehensive evaluation and comparison of enhanced learning methods. *Academic Journal of Science and Technology, 10*(3), 167-171.
11. Dai, Shuying, et al. (2024). The cloud-based design of unmanned constant temperature food delivery trolley in the context of artificial intelligence. *Journal of Computer Technology and Applied Mathematics, 1*(1), 6-12.
12. Liu, Tianrui, et al. (2024). Spam detection and classification based on distilbert deep learning algorithm. *Applied Science and Engineering Journal for Advanced Research, 3*(3), 6-10.
13. Mo, Yuhong, et al. (2024). Make scale invariant feature transform "Fly" with CUDA. *International Journal of Engineering and Management Research, 14*(3), 38-45.
14. He, Shuyao, et al. (2024). Lidar and monocular sensor fusion depth estimation. *Applied Science and Engineering Journal for Advanced Research, 3*(3), 20-26.
15. Samir Elhedhli, Zichao Li, James, & H. Bookbinder. (2017). Airfreight forwarding under system-wide and double discounts. *EURO Journal on Transportation and Logistics, 6*(2), 165–83. https://doi.org/10.1007/s13676-015-0093-5.

16. Liu, Jihang, et al. (2024). Unraveling large language models: From evolution to ethical implications-introduction to large language models. *World Scientific Research Journal*, *10*(5), 97-102.
17. Lin, Zheng, et al. (2024). Text sentiment detection and classification based on integrated learning algorithm. *Applied Science and Engineering Journal for Advanced Research*, *3*(3), 27-33.
18. Zhao, Peng, et al. (2024). Task allocation planning based on hierarchical task network for national economic mobilization. *Journal of Artificial Intelligence General Science*, *5*(1), 22-31.
19. Zhu, Armando, et al. (2024). Cross-task multi-branch vision transformer for facial expression and mask wearing classification. arXiv preprint arXiv:2404.14606.
20. Li, Keqin, et al. (2024). Utilizing deep learning to optimize software development processes. arXiv preprint arXiv:2404.13630.
21. Li, Keqin, et al. (2024). The application of augmented reality (ar) in remote work and education. arXiv preprint arXiv:2404.10579.
22. Wang, Jin, et al. (2024). Research on emotionally intelligent dialogue generation based on automatic dialogue system. arXiv preprint arXiv:2404.11447.
23. C. Zhou, Y. Zhao, Y. Zou, J. Cao, W. Fan, Y. Zhao, & C. Chiyu. (2024 May). *Predict click-through rates with deep interest network model in e-commerce advertising*.
24. C. Zhou, Y. Zhao, S. Liu, Y. Zhao, X. Li, & C. Cheng. (2024). Research on driver facial fatigue detection based on yolov8 model. *ResearchGate*.