Learning Neural Orientation Field for Volumetric Hair Reconstruction

Project Milestone

Fangjun Zhou fzhou48

Weiran Xu weiran

Zhenyu Zhang zhenyuz5

Saturday 9th November, 2024

1 Introduction

Reconstructing human hair is one of the most challenging yet critical process in rendering photorealistic digital human. Unlike other parts of the human body, human hair is highly detailed and often intertwined together. Therefore, it's difficult to use traditional photogrammetry method to reconstruct its structure.

Before machine learning model is used in this field, artists often hand crafted splines on skulls to represent hair strands. Each strand is then textured and rendered to mimic the hair volume. This workflow requires a lot of experience as it's non-trivial for artists to infer the final render result from hair stand splines. To reduce the workload and improve the accuracy of hair reconstruction, machine learning models are trained to generate hair strand from captured images.

In this work, we propose a new method capturing the hair structure by a 3D vector field from $\mathbb{R}^3 \to \mathbb{R}^3$ representing the hair growing direction and occupancy. This mapping can be used later to generate hair stand directly by numerically solving the PDE. It can also be used as a latent variable to guide other generative models as mentioned in [2]. We can fit this vector field by a simple MLP. We then project the vector field onto multiple camera view with a volumetric renderer. Since the screen space projection and volumetric rendering algorithm are differentiable, we are able to learn the vector field from images from multiple camera angle.

2 Related Work

Previous attempt to achieve this goal mainly focus on learning based hair strand generation. This includes some studies about single view hair synthesis [6, 9, 8, 1]. Since the image only contains hair structure from one viewing angle, it's impossible to reconstruct entire hair accurately. These models often use pretrained image encoders such as ResNet-50 [6] to encode the abstract hair style into a feature vector, then use generative models such as U-Net [9], VAE [6], and diffusion models [7] to generate the final strand. These models also struggle with generating curly hair as there's only limited information about growing direction after feature extraction.

In [7] and [5], the authors also tried hair syntheses from multi-view images. However, these two studies still failed to capture finer detail.

Another study about this topic tried to tackle this problem by expanding the traditional PatchMatch MVS (PMVS) algorithm to a Line-based PatchMatch MVS (LPMVS) [4]. This method, despite its high accuracy, doesn't capture the volumetric property of human hair.

Our work is highly inspired by NeRF [3], a model used for 3D reconstruction from 2D images. However, our model differs in two major way.

In NeRF, the model fits a radiance field from $\mathbb{R}^5 \to \mathbb{R}^4$. The input of the randiance function includes the sample position and view direction. However, since the vector field our model fits is view-independent, the input space is only $\mathbb{R}^5$. This makes it easier for the model to capture more information from a smaller dataset.

On top of that, the volumetric renderer on vector field also differs from the one on radiance field. In this case, NeRF only integrate the sampled radiance for each ray, while our model need to project the sample vector onto the filming plane an then integrate the projected vector.

3 Method

3.1 Data Pre-process

Our model will be trained on images of human bust from multiple viewing angles. The preprocessing kernel will extract the projection of the hair growing direction onto the camera viewing plane. This can be done by a simple edge detection kernel such as Gabor filter used in related studies. In this paper, we utilize the preprocessing pipeline in HairStep [9] to segment out the hair from the image and get the direction vector of the top layer hair.

3.2 Backbone

The backbone network is similar to NeRF [3]'s backbone, primarily consisting of multiple MLP layers and residual connections. The difference is that the input to the network we constructed is the spatial coordinates (x, y, z) of the point to be evaluated, and the output is the direction vector of the hair at that point $v = (v_x, v_y, v_z)$, as well as the opacity parameter σ. The direction vector is used to describe the growing direction of the hair at this point, while the opacity parameter sigma is used in loss construction to model the occlusion relationships of the hair which will be discussed later.

3.3 Loss Defination

The construction of the model's loss is divided into two parts: occlusion relationship modeling and plane projection of the direction vector. In occlusion relationship modeling, the weighted 3D direction vector is obtained by integrating the direction vectors along the observation direction ray, weighted by the opacity parameter σ and an function describe the accumulative transmittance, similar to NeRF [3]. We can simulate the integration process through sampling specific points on the observation direction ray and discrete the original integral.

For the second part, we project the weighted 3D direction vector obtained from the first part along the viewing direction onto the camera plane, resulting in the 2D direction vector observed by the camera. The final loss function is constructed by calculating the MSE loss between the 2D direction vector and the ground truth. We will use mini-batched gradient descent to optimize the loss function.

4 Experiment

Given a set of images of human busts from multiple viewing angles, we will first use COLMAP, a general-purpose MVS pipeline that processes a set of images to generate a point cloud, as our baseline. Next, we will evaluate the quality enhancements provided by our model compared to both the Line-based PatchMatch MVS [4] and generative models. Since the final output of our model differs from that of [4] and the generative models, we will not compare their performances directly. Instead, we will evaluate performance relative to COLMAP. Finally, we will demonstrate the tunability of our model's output using Blender.

References

[1] Chongyang Ma. "Single-View Hair Modeling Using A Hairstyle Database". In: ().

[2] Gal Metzer et al. *Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures.* Nov. 14, 2022. arXiv: `2211.07600`. URL: `http://arxiv.org/abs/2211.07600` (visited on 11/09/2024).

[3] Ben Mildenhall et al. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.* Aug. 3, 2020. DOI: `10.48550/arXiv.2003.08934`. arXiv: `2003.08934[cs]`. URL: `http://arxiv.org/abs/2003.08934` (visited on 10/06/2024).

[4] Giljoo Nam et al. "Strand-Accurate Multi-View Hair Capture". In: ().

[5] Radu Alexandru Rosu et al. *Neural Strands: Learning Hair Geometry and Appearance from Multi-View Images.* July 28, 2022. arXiv: `2207.14067[cs]`. URL: `http://arxiv.org/abs/2207.14067` (visited on 10/06/2024).

[6] Shunsuke Saito et al. "3D hair synthesis using volumetric variational autoencoders". In: *ACM Transactions on Graphics* 37.6 (Dec. 31, 2018), pp. 1–12. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3272127.3275019. URL: https://dl.acm.org/doi/10.1145/3272127.3275019 (visited on 10/06/2024).

[7] Vanessa Sklyarova et al. *Neural Haircut: Prior-Guided Strand-Based Hair Reconstruction.* June 12, 2023. DOI: 10.48550/arXiv.2306.05872. arXiv: 2306.05872[cs]. URL: http://arxiv.org/abs/2306.05872 (visited on 10/06/2024).

[8] Keyu Wu et al. "NeuralHDHair: Automatic High-fidelity Hair Modeling from a Single Image Using Implicit Neural Representations". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, June 2022, pp. 1516–1525. ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.00158. URL: https://ieeexplore.ieee.org/document/9878513/ (visited on 10/06/2024).

[9] Yujian Zheng et al. *HairStep: Transfer Synthetic to Real Using Strand and Depth Maps for Single-View 3D Hair Modeling.* Mar. 23, 2023. DOI: 10.48550/arXiv.2303.02700. arXiv: 2303.02700[cs]. URL: http://arxiv.org/abs/2303.02700 (visited on 10/06/2024).