Learning Neural Orientation Field for Volumetric Hair Reconstruction

Project Milestone

Fangjun Zhou fzhou48
Weiran Xu weiran
Zhenyu Zhang zhenyuz5

Thursday 5th December, 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 orientation field from $\mathbb{R}^3 \to \mathbb{R}^2$ representing the hair growing direction and occupancy. On top of that, another occupancy field from $\mathbb{R}^3 \to \mathbb{R}^3$ is learned to indicate void, body, or hair occupancy. These mappings 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 orientation field by a simple MLP. We then project the orientation 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 orientation 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 $\mathbb{R}^5 \rightarrow \mathbb{R}^4$. The input of the randiance function includes the sample position and view direction. However, since the orientation 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 orientation 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 orientation onto the filming plane an then integrate the projected orientation.

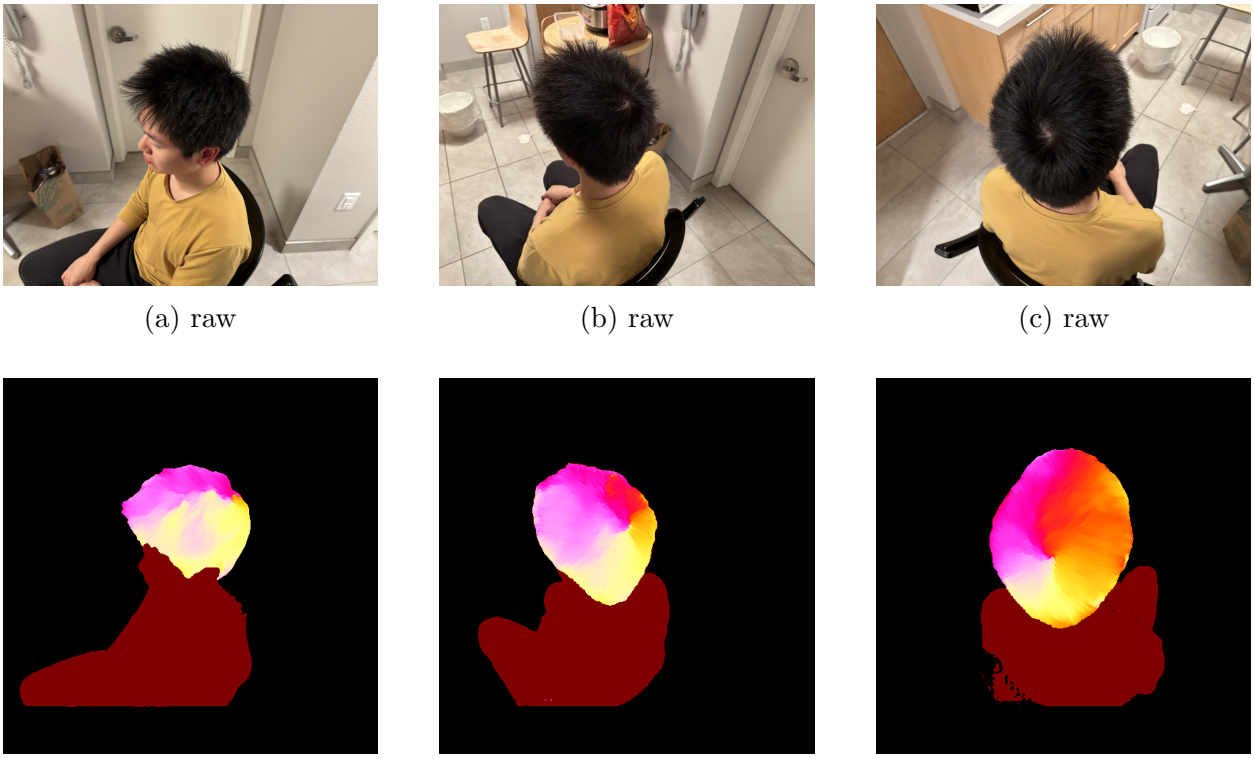
3 Method

3.1 Data Preprocess

Our model will be trained on a set of images of human bust from multiple viewing angles.

Image : For each of the images, the preprocessing kernel will extract the projection of the hair growing direction onto the camera viewing plane. In this paper, we utilize the preprocessing pipeline in HairStep [9] to segment out the hair from the image and get the direction vector field of the top layer hair. Each pixel in the preprocessed image has three channels rgb, we use r channel = 255 to denote the pixel belongs to the hair region, and the g and b channel together as the direction vector. For the region that does not belongs to hair but belongs to human body, its rgb = (127, 0, 0), and for other region we set rgb to $\mathbf{0}$.

Camera : For portrait taken in reality, we use COLMAP, a general-purpose MVS pipeline that processes a set of images to generate a point cloud, to reconstruct the camera position and angle. We also implemented a visualizer for COLMAP for debug purposes. In Figure 2, we see that COLMAP successfully reconstruct camera for the image.



(a) raw (b) raw (c) raw

(d) processed (e) processed (f) processed

Figure 1: Image Preprocessing

3.2 Neural Orientation Field Model

The neural orientation field model we proposed consists of multiple MLP layers and residual connections. The input of the network is the spatial coordinates (x, y, z) of the point to be sampled. The output is the normalized orientation of the hair (θ, ϕ), as well as the opacity parameter $\sigma = (\sigma_{void}, \sigma_{hair}, \sigma_{body})$. A softmax layer is used to make $\|\sigma\|_2 = 1$.

3.3 Volumetric Rendering of a Neural Orientation Field

For each camera in the scene, we can emit camera rays for each pixel on the filming plane. Then, hair orientation and occupancy can be sampled along the camera ray. Given the camera view matrix $V \in \mathbb{R}_{4x4}$, the normalized orientation vector in homogeneous coordinate $\mathbf{o} \in \mathbb{R}^4$ can be projected to the view space by $\mathbf{o}_{view} = V\mathbf{o}$.

To project the orientation vectors onto screen space, we need to apply camera projection to the view space vectors. However, as we choose pinhole camera as our camera model, the projected vectors are independent of the camera parameters. In this cases, the screen space projection of the view space orientation is $\mathbf{o}_{screen}$

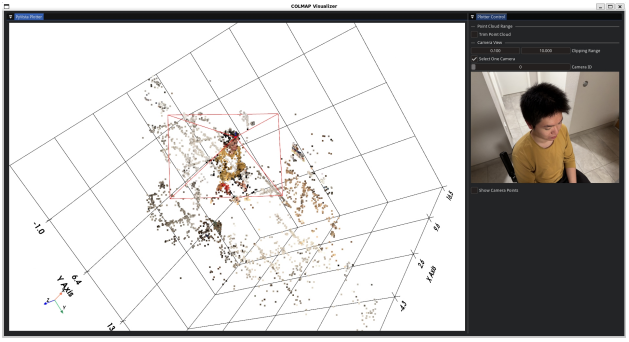


Figure 2: Camera position reconstruction using COLMAP

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

$$\mathbf{o}_{screen} = P\mathbf{o}_{view} \tag{2}$$

Similar to the volumetric rendering function defined in NeRF [3], our rendering function for screen space orientation can be written as

$$O(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma_{hair}(\mathbf{r}(t))PV\mathbf{o}(\mathbf{r}(t), \mathbf{d})dt \tag{3}$$

$$T(t) = exp(-\int_{t_n}^{t_f} \sigma_{hair}(\mathbf{r}(s))ds) \tag{4}$$

One important difference between our model and NeRF is we use two separate occupancy for body and hair, while in NeRF one occupancy is used to integrate camera rays. In our case, camera rays can be blocked by face and body that doesn't contribute to the final hair orientation field. These parts are marked red in the preprocessing step s demonstrated in Figure 1. When training the model, we integrate the 2D hair orientation using the aforementioned volumetric render function. We also run a traditional NeRF volumetric renderer to integrate the body occupancy to generate the body mask. The final loss is defined as the combination of two losses.

4 Experiment

4.1 DataSet

To investigate the convergence behavior of NeRF, we utilized synthetic images generated from Blender, specifically using a scene of Hoover Tower. For hair direction reconstruction,

we compiled a dataset of 41 portraits, captured by ourselves, featuring one of our group members.

4.2 Metrics

The Peak Signal-to-Noise Ratio (PSNR) is a widely used metric in signal processing for assessing image similarity, particularly in image reconstruction tasks. It effectively represents the Mean Squared Error (MSE) between two images in a 2D context.

4.3 Baseline

We attempted to train a NeRF model to represent hair structure as a radiance field, leveraging it to infer hair orientation. Since NeRF does not inherently encode orientation information, we relied on HairStep to predict orientation. Initial observations showed a slow convergence rate when training on highly detailed scenes, consistent with findings in the original NeRF paper [3], which noted that NeRF tends to capture low-frequency features early in training, see Figure 3. Another consequence of this is that the output of NeRF is often blurred, making HairStep difficult to segment out hair and output reasonable hair direction, thus incapable of holding relevant direction information, see Figure 4.

(a) step 800 (b) step 4000 (c) step 10000

Figure 3: NeRF's output on evaluation set, shows slow convergence and high variance

We also observed signs of overfitting in NeRF, with the PSNR on the evaluation set stabilizing around 1000 batches. We aim to address this with regularization methods (WIP). Additionally, we hypothesize that the model may converge faster on a smaller dataset, given that the hair orientation field we aim to fit is view-independent and should not vary with camera position. As the outcome is evident, we opted not to measure PSNR for the NeRF baseline.

(a) raw

(b) NeRF output

(c) hair direction extraction ground truth

(d) hair direction extraction from NeRF output

Figure 4: Baseline Experiment

5 Contribution

5.1 Fangjun Zhou

Wrote preprocessing script that uses COLMAP to extract features and camera poses. Wrote colmap_visualizer using imagui and pyvista to help visualize the extracted point cloud / camera position. Wrote helper methods to construct camera rays for NeRF training. Rendered hoover tower dataset in Blender. Implement and trained NeRF baseline. Derive orientation vector projection volumetric render function. Wrote method section in the milestone report.

5.2 Weiran Xu

Contributed to building the human portrait dataset and applied HairStep for image preprocessing and hair strand direction extraction, preparing the data for analysis. Also wrote part of experiment section in the milestone report, detailing methodology and preliminary findings.

5.3 Zhenyu Zhang

Responsible for refactoring Tiny NeRF, do the training and testing related to Tiny NeRF, and implementing the code for 3D vector field reconstruction of hair growth. This included model code, loss computation code, camera view transformation and coordinate system conversion code, as well as the code of calculations for remapping 3D vectors in space to the camera frame.

6 Work in Progress

We've finish implementing our proposed model but haven't start the experiment. We're also planning to add more baselines to compare with. On top of that, the current model is slow to converge. We're planning to implement hierarchical volume sampling mentioned in the original NeRF paper to speed up convergence.

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).