

Learning Neural Orientation Field for Volumetric Hair Reconstruction

Fangjun Zhou¹ Zhenyu Zhang¹ Weiran Xu¹

¹Computer Science, Stanford



Introduction

Reconstructing human hair is one of the most challenging yet critical process in rendering photorealistic digital human. In this work, we propose a new method capturing the hair structure by learning a 3D orientation field from $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ representing the hair growing direction. Given a set of portraits with annotated hair orientation, our method effectively reconstructs hair orientation from unseen angles with high-resolution and robustness.

Data

The dataset used by this project is generated by a Blender geometry node-based hair system demo file. We render the model from multiple viewing angles and exported the camera intrinsics and extrinsics for later use.

Our training set consists of 128 instances and our test set consists of 16 instances, all rendered from the same model but with different camera poses.



Figure 1. Synthetic dataset rendered by Blender

The Neural Orientation Field (NeOF) Model

The neural orientation field (NeOF) model we propose consists of multiple MLP layers and residual connections.

- **Input** Spatial coordinates (x, y, z) of the point to be sampled
- **Output** World space hair orientation (θ, ϕ) , and opacity parameter $\sigma = (\sigma_{hair}, \sigma_{body})$

Volumetric Rendering for Neural Orientation Field

We propose a differentiable volumetric renderer to render both screen-space hair orientation, body/hair masks, and occupancy.

To sample a pixel, we emit a camera ray with the known camera intrinsics and extrinsics on the filming plane. (Figure 2) Then, the world space hair orientation and occupancy can be sampled along the camera ray. The volumetric rendering pipeline first projects the world space orientation to screen space, then integrate the screen space orientation with the sampled occupancy.

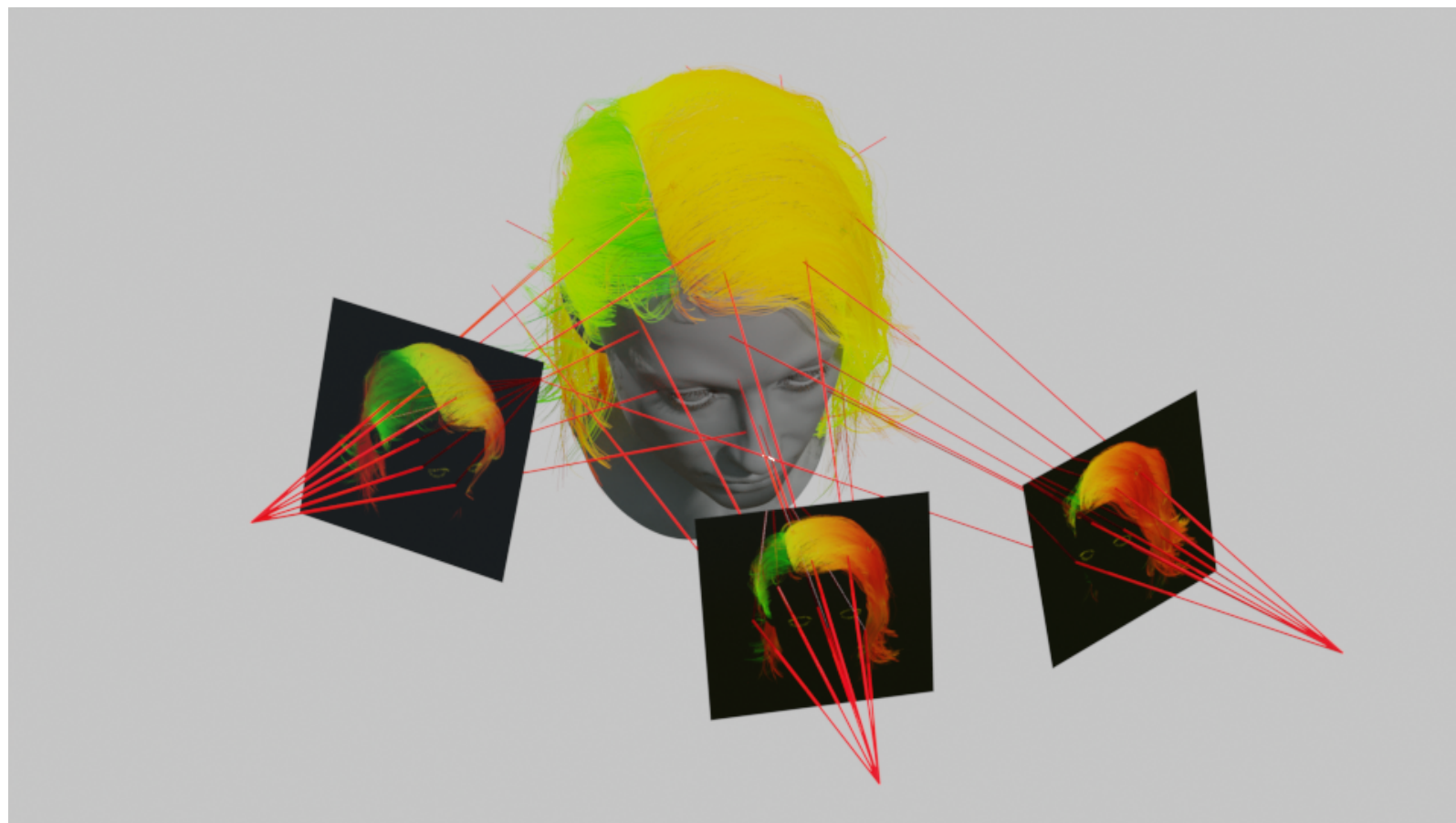


Figure 2. Volumetric Renderer

Volumetric Rendering for Neural Orientation Field

To project hair orientation, we first need to convert the world space orientation \mathbf{o}_{world} to an orientation vector in homogeneous coordinate $\mathbf{v}_{world} \in \mathbb{R}^4$. Given the camera extrinsic matrix $M \in \mathbb{R}^{4 \times 4}$, \mathbf{v}_{world} can be projected to the view space by $\mathbf{v}_{view} = M\mathbf{v}_{world}$.

Normally, to project a world space vector onto screen space, we need to first apply view transform to get view space projection, then apply perspective transform to get screen space projection. However, as we choose pinhole camera as our camera model, the screen space orientation vectors are independent of the camera intrinsics. In other words, $\mathbf{v}_{screen} = \mathbf{v}_{view} = M\mathbf{v}_{world}$.

Similar to the volumetric rendering function defined in NeRF [1], our rendering function for screen space orientation can be written as:

$$V(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma_{hair}(\mathbf{r}(t)) M \mathbf{v}_{world}(\mathbf{r}(t)) dt \quad (1)$$

$$T(t) = \exp\left(-\int_{t_n}^t \sigma_{body}(\mathbf{r}(s)) ds\right) \quad (2)$$

One important difference between our model and NeRF is we use two separate occupancy for body and hair, while in NeRF only 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 integration. Therefore, we use σ_{body} instead of σ_{hair} to integrate residual ray $T(t)$.

As we provide body mask and hair mask in the training data, we can also use the following rendering function to render body mask $B(\mathbf{r})$ and hair mask $H(\mathbf{r})$:

$$B(\mathbf{r}) = 1 - \exp\left(-\int_{t_n}^{t_f} \sigma_{body}(\mathbf{r}(s)) ds\right) \quad (3)$$

$$H(\mathbf{r}) = 1 - \exp\left(-\int_{t_n}^{t_f} \sigma_{hair}(\mathbf{r}(s)) ds\right) \quad (4)$$

Similar to the numerical estimation of radiance integration mentioned in NeRF, we use numerical estimation of orientation and mask integration defined as follows:

$$\hat{V}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_{hair}^{(i)} \delta_i)) M \mathbf{v}_{world}^{(i)} \quad (5)$$

$$T_i = \exp\left(-\sum_{j=1}^i \sigma_{body}^{(j)} \delta_j\right) \quad (6)$$

$$\hat{B}(\mathbf{r}) = 1 - \exp\left(-\sum_{i=1}^N \sigma_{body}^{(i)} \delta_i\right) \quad (7)$$

$$\hat{H}(\mathbf{r}) = 1 - \exp\left(-\sum_{i=1}^N \sigma_{hair}^{(i)} \delta_i\right) \quad (8)$$

where $\delta_i = t_{i+1} - t_i$. $\hat{V}(\mathbf{r})$, $\hat{B}(\mathbf{r})$, and $\hat{H}(\mathbf{r})$ are the final outputs of our volumetric render. The final loss is defined as the sum of orientation loss and mask losses. In theory, only orientation loss is required for the model to converge. However, we observed that introducing the mask losses helps improving the convergence speed. It also improves the model's performance on occupancy prediction accuracy.

Result and Discussion

By comparing our method with the two baselines, HairStep [2] and NeRF, we can observe from the evaluation metrics PSNR and MSE that our results outperform the two baselines. The result shows in Table 1. Sample images are shown in Figure 3. We represent the direction of planar hair at a given point using vectors formed by the red and green channels of the RGB images.

Method	PSNR	MSE
NeOF (ours)	15.06 dB	2026.75
HairStep	13.97 dB	2608.00
NeRF	10.05 dB	6429.77

Table 1. Comparison of losses between different methods

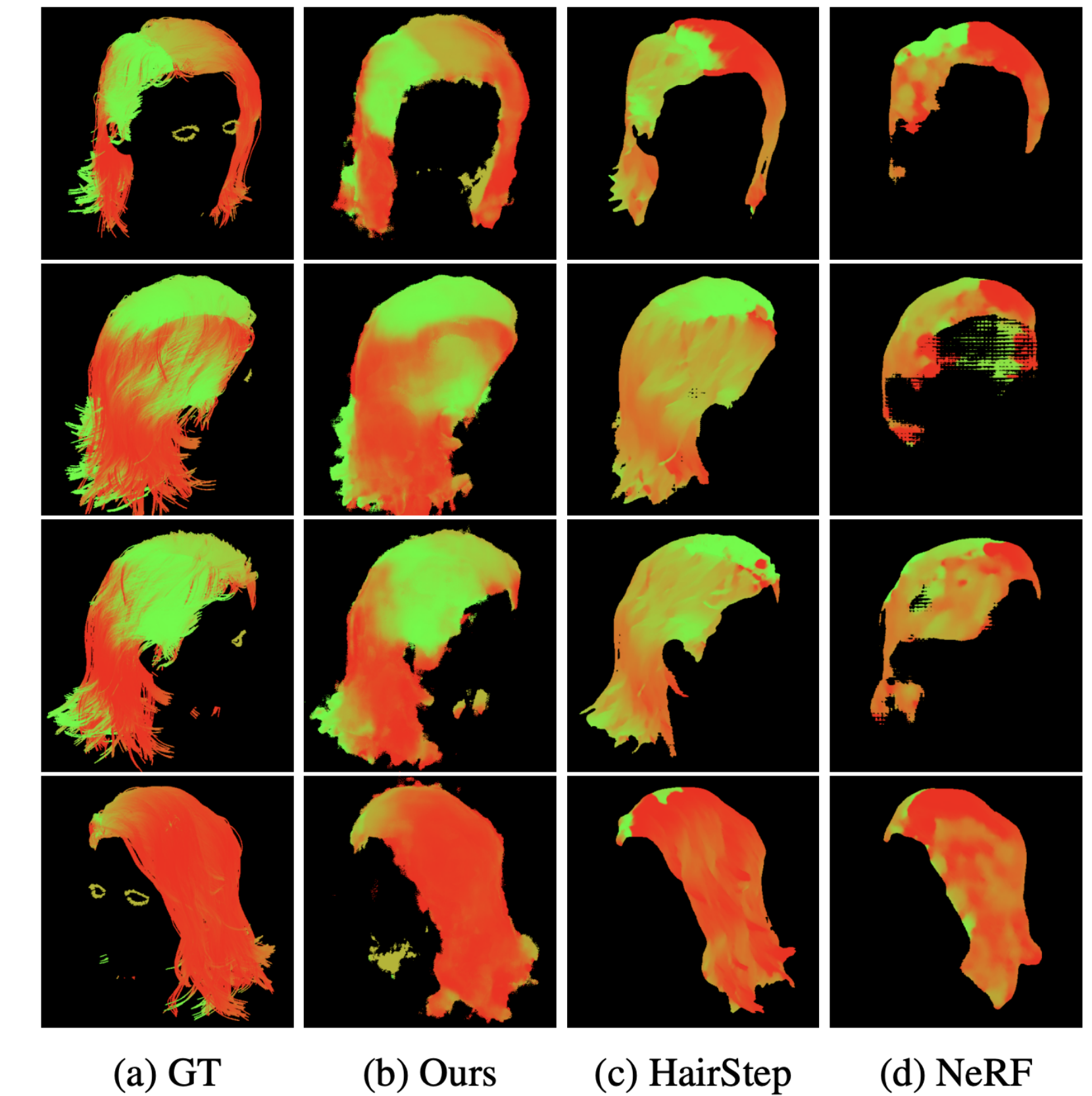


Figure 3. Normalized hair orientations, where (a) is the Ground Truth, (b) is the result our NeOF method, (c) is result of HairStep, and (d) is the result of traditional NeRF with HairStep.

The overall vector field distribution of our method closely matches the ground truth. However, HairStep makes some errors when predicting some unintended horizontal components in areas where the hair extends vertically. This leads to discrepancies with the ground truth. For NeRF, the reconstructed 3D model is so blurry that hardly tells the strands apart, leading large errors in predicting the hair growth directions.

Future

In the future, we aim to realize a complete 3D reconstruction pipeline, enabling our method to directly output editable 3D models compatible with Blender, facilitating further editing. We believe this approach will provide a straightforward and efficient solution for sampling and automating the modeling of real human hair, significantly reducing the costs associated with traditional hair modeling in the visual effects and animation industries.

References

- [1] 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.
- [2] Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. HairStep: Transfer synthetic to real using strand and depth maps for single-view 3d hair modeling.