Neural Haircut

Prior-Guided Strand-Based Hair Reconstruction

Ekaterina Kozlova

https://samsunglabs.github.io/NeuralHaircut/static/videos/jenya_teaser3.mp4

Overview

- hair is a set of 3D-polylines (**strands**), strand is a set of points
- [authors] propose method for hair modeling that uses **only image- or video-based data without any additional manual annotations**
- two-stage reconstruction pipeline:
- obtain a **coarse volumetric hair reconstruction** in the form of **implicit fields**
- reconstruct **fine hair strands** using optimization of coarse geometry-based, rendering-based, and prior-based terms
- **hairstyle prior** is obtained separately during **pretraining on a synthetic dataset**

Hair Prior: motivation and overview

- **motivation: hairstyle priors ensure the physical realism of the reconstructed strands**
- global hairstyle prior is trained using a **strand parametric model** and a **latent diffusion network**
- they interface with each other via the **geometry texture** T
- we use **synthetic datasets** while training the prior, so strands (their points) are known

Hair Prior: Hair Strand Parametric Model

- **VAE**: mapping strand $S = \{p^l\}_{l=1}^{L}$ to a **latent vector** z (we need them for the diffusion model)
- **reparametization**: $z = z_\mu + \varepsilon \cdot z_\sigma, \varepsilon \sim \mathcal{N}(0,1)$, z_μ, z_σ are the parameters of the Gaussian distribution
- data term: compare **points**, their **orientations**, and **curvatures** of strands
- data term improves the **fidelity of curly hair** reconstructions
- $$\mathcal{L}_{\text{data}} = \sum_{l=1}^{L} \left\| \hat{p}^l - p^l \right\|_2^2 + \lambda_d \left(1 - \hat{b}^l b^l\right) + \lambda_c \left\| \hat{g}^l - g^l \right\|_2^2$$
- KL divergence is estimated between $\mathcal{N}(z_\mu, z_\sigma)$ and $\mathcal{N}(0,I)$
- $\mathcal{L}_{VAE} = \mathcal{L}_{\text{data}} + \lambda_{KL}\mathcal{L}_{KL}$

Hair Prior: Hairstyle Diffusion Model

- hairstyle = N strands $\{S_i\}_{i=1}^{N}$ => N latent vectors $\{z_i\}_{i=1}^{N}$, use pretrained encoder
- convert the vectors into dense texture T using **nearest neighbour interpolation**
- **dense texture is a two-dimensional map of latent hair vectors, where the position on the map corresponds to the position of the hair root on the scalp**
- apply augmentations to texture to increase the diversity of training samples
- the texture is then subsampled into a **low-resolution map** T_{LR}
- denoiser $\mathscr{D}$ is the **Elucidating Diffusion Model** (EDM, for some reason EDM > DDPM)
- $y = \mathrm{T}_{LR}, x = y + \varepsilon \cdot \sigma, \varepsilon \sim \mathcal{N}(0,1), \sigma$ is a noise strength
- denoising process: $\mathscr{D}(\mathbf{x}, \sigma) = c_{\text{skip}}(\sigma) \cdot \mathbf{x} + c_{\text{out}}(\sigma) \cdot \mathscr{F}\left(c_{\text{in}}(\sigma) \cdot \mathbf{x}, c_{\text{noise}}(\sigma)\right)$

see why EDM > DDPM: https://arxiv.org/pdf/2206.00364.pdf

Hair Prior: Hairstyle Diffusion Model

- denoising: $\mathscr{D}(\mathbf{x}, \sigma) = c_{\text{skip}}(\sigma) \cdot \mathbf{x} + c_{\text{out}}(\sigma) \cdot \mathscr{F}\left(c_{\text{in}}(\sigma) \cdot \mathbf{x}, c_{\text{noise}}(\sigma)\right)$
- $\mathscr{F}$ is the neural network to be trained, $c_{\text{skip}}(\sigma)$ modulates the skip connection, $c_{\text{in}}(\sigma)$ and $c_{\text{out}}(\sigma)$ scale the input and output magnitudes, and $c_{\text{noise}}(\sigma)$ maps noise level into a conditioning input for $\mathscr{F}$
- $\mathscr{L}_{\text{diff}} = \mathbb{E}_{\mathbf{y},\sigma,\epsilon}\left[\lambda_{\text{diff}}(\sigma) \cdot \|\mathscr{D}(\mathbf{x}, \sigma) - \mathbf{y}\|_2^2\right]$
- $\lambda_{\text{diff}}(\sigma)$ is a weighting function, and the expectation is approximated via sampling
- **motivation**: we want to add a regularization term $\mathscr{L}_{\text{prior}}$ during fine strand-based reconstruction to enhance geometry texture of the generated hairstyle

Coarse Volumetric Reconstruction

Multi-view images

Stage I: Volumetric hair reconstruction

Coarse Volumetric Reconstruction: overview

- during the first stage, we reconstruct **implicit surface representations** for hair and bust (head and shoulders) regions
- additionally, we learn a **field of hair growth directions**, which we call 3D orientations, by matching them through a differentiable projection with hair directions observed in the training images or 2D orientation maps
- its primary use case is to **constrain the optimization of hair strands during the second stage**
- to calculate the hair orientation maps from the input frames, we use a classic approach based on image gradients

https://lingjie0206.github.io/papers/NeuS/

Coarse Volumetric Reconstruction

Neus + SDF

- use only image- and video-based data, and have no information about the strands
- coarse reconstruction is made by estimating hair and bust geometry as **signed distance functions** $f_{hair}, f_{bust} : \mathbb{R}^3 \rightarrow \mathbb{R}$
- SDFs: some function that maps spatial position to its signed distance to the object
- volumetric **ray marching approach** for neural implicit **surfaces**, NeuS, is used to fit SDFs; hair geometry and bust geometry are **separate shapes**
- NeRF is a volume rendering based methods, NeuS combines the advantages of surface rendering based and volume rendering based methods
- NeuS ~= 3 MLPs (for hair, for bust, for scene color)

https://lingjie0206.github.io/papers/NeuS/

Coarse Volumetric Reconstruction

Neus + SDF

- color prediction: approximating a pixel's color $\mathbf{c}$ using the radiance at N points $\mathbf{x}_i$ sampled along the corresponding ray $\mathbf{v}$
- $$\hat{\mathbf{c}} = \sum_{i=1}^{N} T_i \cdot \alpha_i \cdot c(\mathbf{x}_i, \mathbf{v}, l, \mathbf{n}), \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$
- T_i is the accumulated transmittance, α_i is the opacity, l and $\mathbf{n}$ - the blended hair with bust features and normals correspondingly, and c is the view-dependent radiance field (captures information about how light interacts with surfaces and materials)
- opacity α_i of each point along the ray is calculated by blending the individual opacities of hair and bust: $\alpha_i = \min\left(\alpha_i^{\text{hair}} + \alpha_i^{\text{bust}}, 1\right)$

Coarse Volumetric Reconstruction

Neus + SDF

- opacity α_i of each point along the ray is calculated by blending the individual opacities of hair and bust: $\alpha_i = \min\left(\alpha_i^{\text{hair}} + \alpha_i^{\text{bust}}, 1\right)$

- render the bust and the hair masks:

$$\hat{\mathbf{o}}_{\text{hair}} = \sum_{i=1}^{N} T_i \cdot \alpha_i^{\text{hair}}, \quad \hat{\mathbf{o}}_{\text{bust}} = \sum_{i=1}^{N} T_i \cdot \alpha_i^{\text{bust}}$$

- training losses include a photometric L1 loss $\mathscr{L}_{\text{color}}$, which matches $\hat{\mathbf{c}}$ and $\mathbf{c}$, a mask-based loss $\mathscr{L}_{\text{mask}}$ that applies binary cross-entropy between the predicted masks and the ground-truth $\mathbf{m}_{\text{hair}}$ and $\mathbf{m}_{\text{bust}}$, and the regularizing Eikonal term $\mathscr{L}_{\text{reg}}$, which is applied for both f_{hair} and f_{bust}

Coarse Volumetric Reconstruction

Head mesh

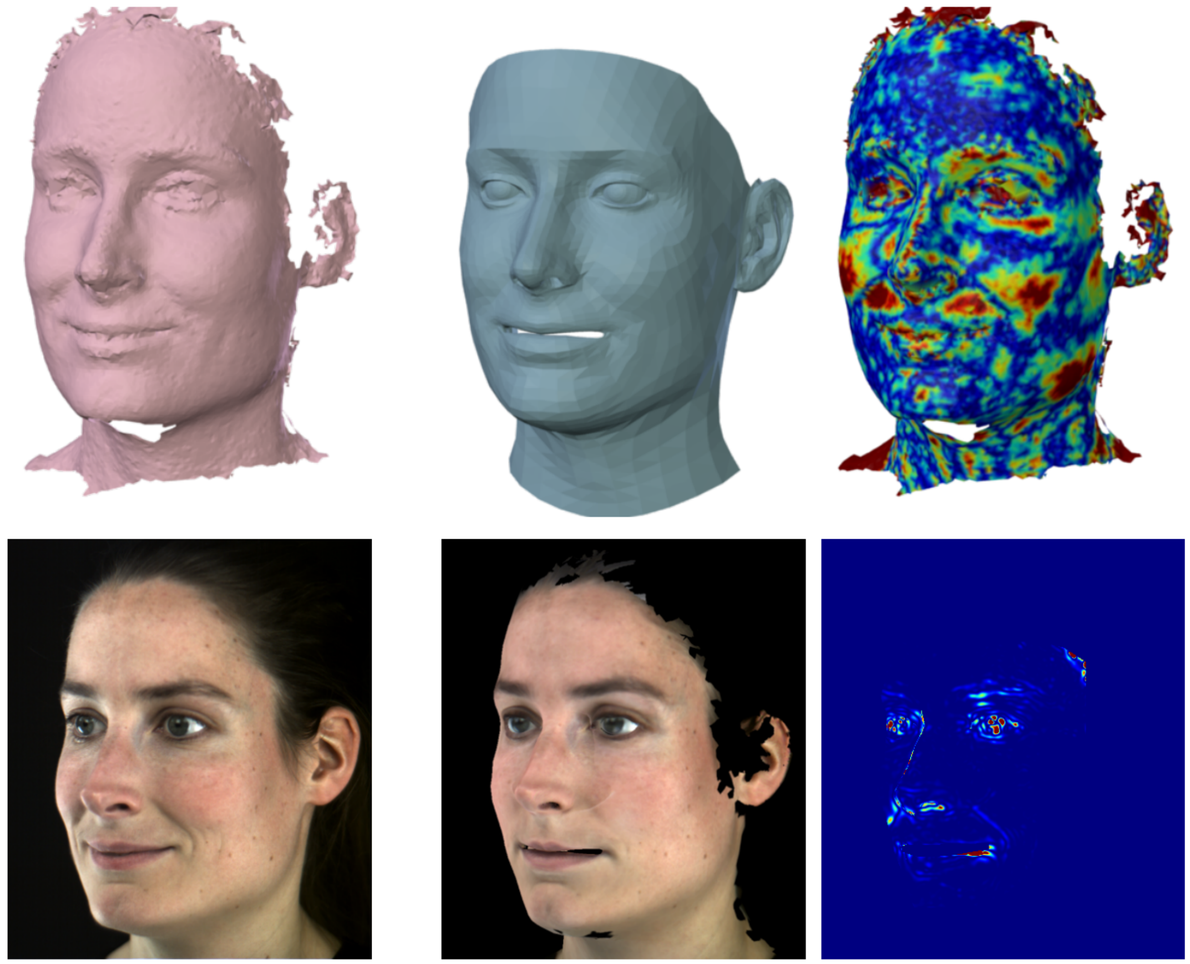
- additional loss $\mathscr{L}_{\text{head}}$: regularization for the bust shape
- fit a FLAME head mesh into the scene using 2D facial landmarks (bust features)
- the mesh is used to ensure that f_{bust} includes the head scalp surface region
- $\mathscr{L}_{\text{head}}$ matches the SDF to the head mesh

Geometry prior: FLAME [Li et al. 2017] is a statistical 3D head model that combines separate linear identity shape and expression spaces with linear blend skinning (LBS) and pose-dependent corrective blendshapes to articulate the neck, jaw, and eyeballs. Given parameters of facial identity $\boldsymbol{\beta} \in \mathbb{R}^{|\boldsymbol{\beta}|}$, pose $\boldsymbol{\theta} \in \mathbb{R}^{3k+3}$ (with $k = 4$ joints for neck, jaw, and eyeballs), and expression $\boldsymbol{\psi} \in \mathbb{R}^{|\boldsymbol{\psi}|}$, FLAME outputs a mesh with $n = 5023$ vertices. The model is defined as

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) = W(T_P(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}), \mathbf{J}(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W}), \quad (1)$$

Coarse Volumetric Reconstruction

Head mesh



scan model-only registration

Coarse Volumetric Reconstruction

Orientation field β

- β: field of hair growth directions, which is obtained via differentiable surface rendering of f_{hair}
- x_s: intersection point of the ray v with the hair surface
- project 3D orientation field $\beta(x_s)$ into the camera $\mathscr{P}$ using Plucker line coords
- projected direction $\mathrm{L}(x_s, \beta(x_s), \mathscr{P})$ is matched to the 2D orientation map, estimated from the training images using Gabor filters
- measure the angle $\hat{a}_{\mathrm{v}}$ between camera's y-axis and predicted direction
- $$\mathscr{L}_{\text{dir}} = \sum_{\mathbf{v}} \frac{\mathbf{m}_{\text{hair}}(\mathbf{v})}{\text{Var}^2\left[a_{\mathbf{v}}\right]} \min\left\{ \left| a_{\mathbf{v}} - \hat{a}_{\mathbf{v}} \right|, \left| a_{\mathbf{v}} - \hat{a}_{\mathbf{v}} \pm \pi \right| \right\}$$

Coarse Volumetric Reconstruction

$$\mathcal{L}_{\text{coarse}} = \mathcal{L}_{\text{color}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{head}}\mathcal{L}_{\text{head}} + \lambda_{\text{dir}}\mathcal{L}_{\text{dir}}$$

Fine Strand-Based Reconstruction: overview

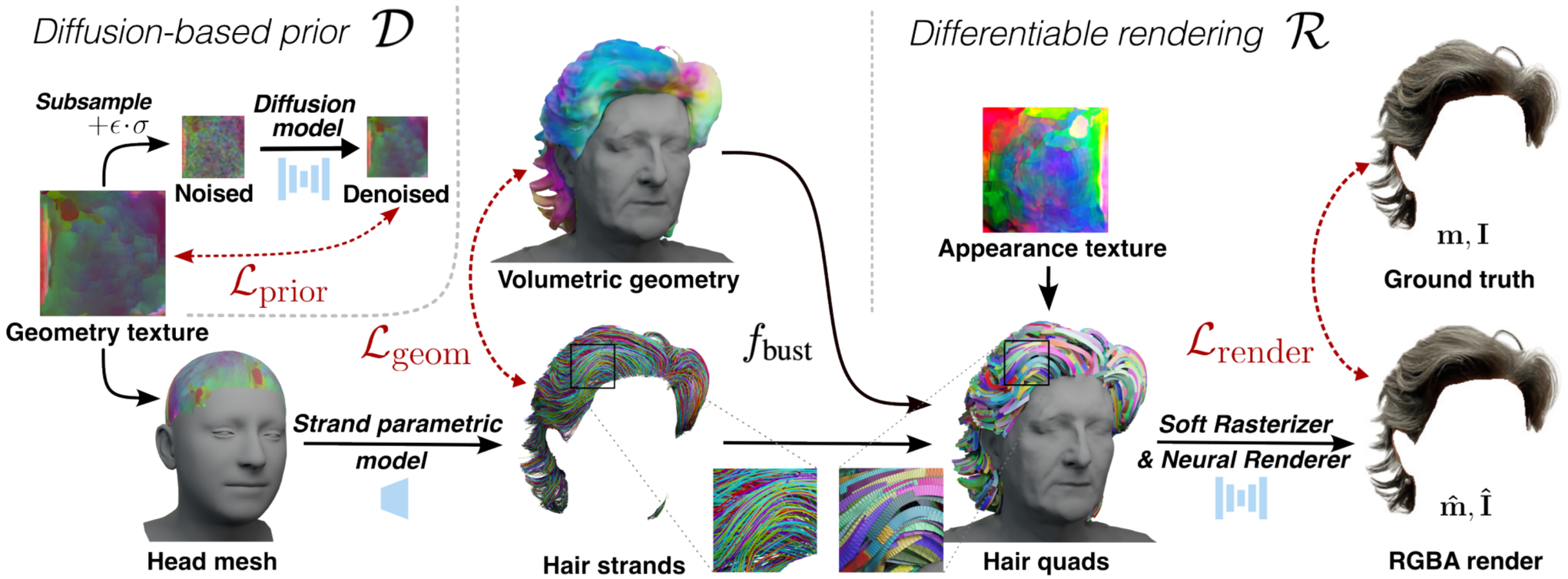


Figure 2: The overview of the second stage of our approach (fine strand-based reconstruction). We use shape texture to represent hair strands and utilize multiple objectives to optimize it. We apply $\mathcal{L}_{\text{prior}}$ as a regularization penalty using a diffusion network pre-trained on synthetic hairstyles. Then, we use $\mathcal{L}_{\text{geom}}$ to match the reconstructed strands to geometry and orientation fields parameterized by the implicit function. Lastly, $\mathcal{L}_{\text{render}}$ is used to match the rendered hair to the ground truth image.

Fine Strand-Based Reconstruction

- **task**: reconstruct hair strands from a geometry texture $\mathbf{T}$
- at each iteration, we sample N random embeddings $\{\mathbf{z}_i\}_{i=1}^{N}$ from the texture $\mathbf{T}$ and obtain corresponding strands $\{\mathbf{S}_i\}_{i=1}^{N}$ using a pre-trained decoder $\mathscr{G}$
- **strands** are then used to evaluate **geometric and rendering-based constraints**
- **prior-based regularization** is applied directly to the geometry texture $\mathbf{T}$ using a **pre-trained diffusion model**

Fine Strand-Based Reconstruction

Geometry-based losses

- $$\mathscr{L}_{\text{vol}} = \sum_{i=1}^{N} \sum_{l=1}^{L} \mathbb{I}\left[f_{\text{hair}}\left(\mathbf{p}_i^l\right) > 0\right] \left(f_{\text{hair}}\left(\mathbf{p}_i^l\right)\right)^2$$
- $\mathscr{L}_{\text{vol}}$: penalizing the points on the strands that stray outside of it
- $$\mathscr{L}_{\text{chm}} = \sum_{k=1}^{K} \left\| \mathbf{x}_k - \mathbf{p}_k \right\|_2^2$$
- $\mathscr{L}_{\text{chm}}$: minimize the distance between K random points $\mathbf{x}_k$ sampled on the coarse hair surface and their nearest points on the strands,
- $$\mathscr{L}_{\text{orient}} = \sum_{m=1}^{M} \left(1 - \left|\mathbf{b}_m \cdot \beta\left(\mathbf{p}_m\right)\right|\right)$$
- $\mathscr{L}_{\text{orient}}$: take points $\mathbf{p}_m$ on strands that are close to the visible hair surface and estimate their orientations $\mathbf{b}_m$, then compare the orientations with the implicit field $\beta(\mathbf{p}_m)$

Fine Strand-Based Reconstruction

Geometry-based losses

$$\mathscr{L}_{\text{geom}} = \mathscr{L}_{\text{vol}} + \lambda_{\text{chm}}\mathscr{L}_{\text{chm}} + \lambda_{\text{orient}}\mathscr{L}_{\text{orient}}$$

Fine Strand-Based Reconstruction

Hair quads

- task: **differentiable rendering** of hair strands

- hair strands → hair quads: stripe-like mesh, which follows the strand trajectory and has normals oriented towards the camera

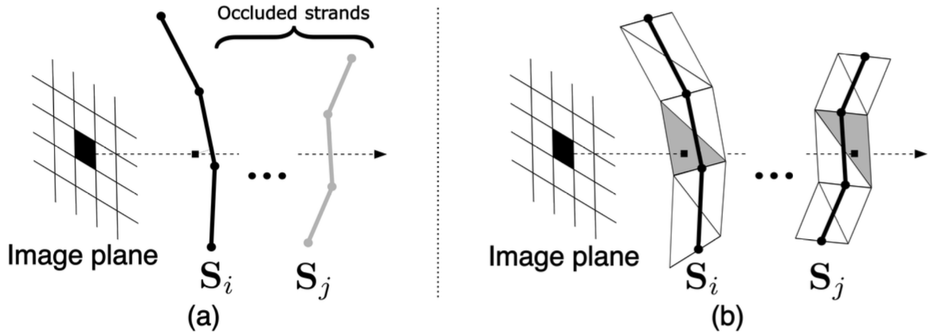


Figure 3: (a) Differentiable hair rasterization algorithm of [58] propagates the gradient only into the first element of z-buffer. (b) Our proposed hair rasterization based on quads leverages soft hair rasterization [36] and passes gradients into multiple elements of the z-buffer to achieve better reconstructions.

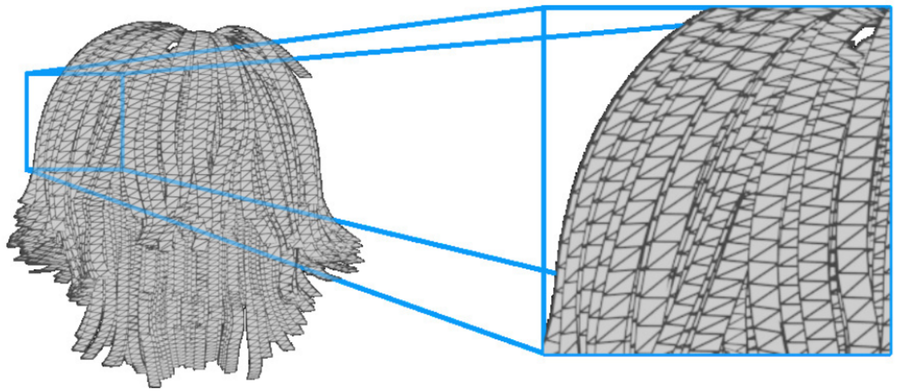


Figure 8: Hair quads produced by the view-aware generation: geometry is built in a way that most of the quads are facing the camera plane.

Fine Strand-Based Reconstruction

Rendering pipeline $\mathscr{R}$

- the vertices of the resulting quad mesh are fully differentiable w.r.t. the strands
- render this mesh using **soft rasterization** => obtain z-buffer (a set of 16 nearest faces to each pixel), blend the faces by using sigmoid
- to render the color, we use a neural rendering approach (**UNet**) that can handle the view-dependent reflectance of the hair
- input: appearance descriptors $\in \mathbb{R}^{16}$ concatenated with rasterized orientations $\hat{a}$
- appearance descriptors are the same for the whole strand, **orientations are different for each point**
- projected orientations contain information about both hair local strand growth direction and camera view direction

Fine Strand-Based Reconstruction

Rendering-based losses

- use neural rendering pipeline $\mathscr{R}_\phi$ to obtain the hair silhouette and the images: $\hat{\mathbf{m}}, \hat{\mathbf{I}} = \mathscr{R}_\phi \left(\{\mathbf{S}_i\}_{i=1}^{N}, f_{\text{bust}}, \mathscr{P} \right)$
- $\mathscr{L}_{\text{mask}}, \mathscr{L}_{\text{rgb}}$: use L1-loss to match the predicted silhouette and the color to the ground truth $\mathbf{m}$ and $\mathbf{I}$
- $\mathscr{L}_{\text{render}} = \mathscr{L}_{\text{rgb}} + \lambda_{\text{mask}} \mathscr{L}_{\text{mask}}$

Diffusion-based prior: SDS

- **score distillation sampling**: the pre-trained diffusion model is used to guide the optimization of a **neural radiance field** by providing it with the gradients in the image space
- these gradients originate from the same loss used to train a diffusion model, in our case, $\mathscr{L}_{\text{diff}}$
- to calculate this loss, we employ **the same procedure as during the training of the diffusion model:**
- sample random noise ε and the noise level σ and apply them to the geometry map
- perform **random sub-sampling** to decrease the resolution of $\mathbf{T}_\theta$ before forwarding it through the diffusion model
- back-propagate the loss $\mathscr{L}_{\text{prior}}$ directly into the parameters θ of the geometry texture $\mathbf{T}_\theta$ while **keeping the weights of the denoiser frozen**

Training the model

- **three days per subject on a single NVIDIA RTX 4090** (one for the first stage, two for the second stage)
- to train a model on real data, we need to **obtain segmentation masks** for the hair and bust (using off-the-shelf methods)
- to **parametrize the geometry texture map** (so we can use SDS) a UNet is used, it predicts the map from a constant mesh grid
- estimation: render the **ground-truth strands using Blender** and reconstruct them
- measure **precision**, **recall**, and **F-score** between our predicted strands and ground truth using both distance and angular errors as thresholds

Ablation study

Method	Straight hair									Curly hair								
	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40
	Precision			**Recall**			**F-score**			**Precision**			**Recall**			**F-score**		
$\mathcal{L}_{\text{geom}}$	63.8	88.6	94.9	9.9	16.2	21.2	17.1	27.4	34.7	50.8	75.1	85.9	5.7	11.3	18.4	10.2	19.6	30.3
w/o $\mathcal{L}_{\text{chm}}$	82.9	95.0	97.1	4.5	8.9	14.2	8.5	16.3	24.8	51.0	73.8	84.6	3.9	8.4	14.3	7.2	15.1	24.5
w/o $\mathcal{L}_{\text{vol}}$	48.3	71.8	79.4	10.1	21.7	32.2	16.7	33.3	45.8	20.1	35.3	45.5	5.7	12.4	21.2	8.9	18.4	28.9
w/o $\mathcal{L}_{\text{orient}}$	31.7	56.2	69.0	6.0	12.1	17.8	10.1	19.9	28.3	21.5	43.7	59.8	4.7	10.3	17.7	7.7	16.7	27.3
w/ $\mathcal{L}_{\text{render}}$ [58]	68.4	89.4	95	9.8	15.7	23.6	17.1	26.7	37.8	48.7	75.3	87.0	6.2	12.0	19.3	11.0	20.7	31.6
w/ $\mathcal{L}_{\text{rgb}}$	71.6	90.4	95.2	9.1	15.6	22.5	16.1	26.6	36.4	49.3	76.0	87.7	6.1	12.0	19.4	10.9	20.7	31.8
w/ $\mathcal{L}_{\text{mask}}$	63.5	88.2	94.6	11.1	17.3	22.5	18.9	28.9	36.4	49.4	74.7	86.1	6.3	12.1	19.5	11.2	12.1	31.8
$\mathcal{L}_{\text{fine}}$ w/o $\mathcal{L}_{\text{rgb}}$	59.8	84.1	92.2	12.9	22.8	31.3	21.2	35.9	46.7	45.1	71.1	83.6	6.3	12.4	20.3	11.1	21.1	32.7
$\mathcal{L}_{\text{fine}}$	59.9	84.1	92.1	13.1	22.7	31.5	21.5	35.8	46.9	45.8	72.1	84.6	6.4	12.8	21.0	11.2	21.7	33.6
Neural Strands* [58]	74.0	81.8	85.3	12.8	20.5	28.8	21.8	32.8	43.1	38.4	59.8	72.4	7.9	15.1	23.8	13.1	24.1	35.8

Table 2: We provide an extended quantitative evaluation of individual components of our method with per-scene metrics. Our full method with $\mathcal{L}_{\text{fine}}$ outperforms others in terms of Recall and F-score for both scenes. For a detailed discussion, please refer to Section B.

Ablation study

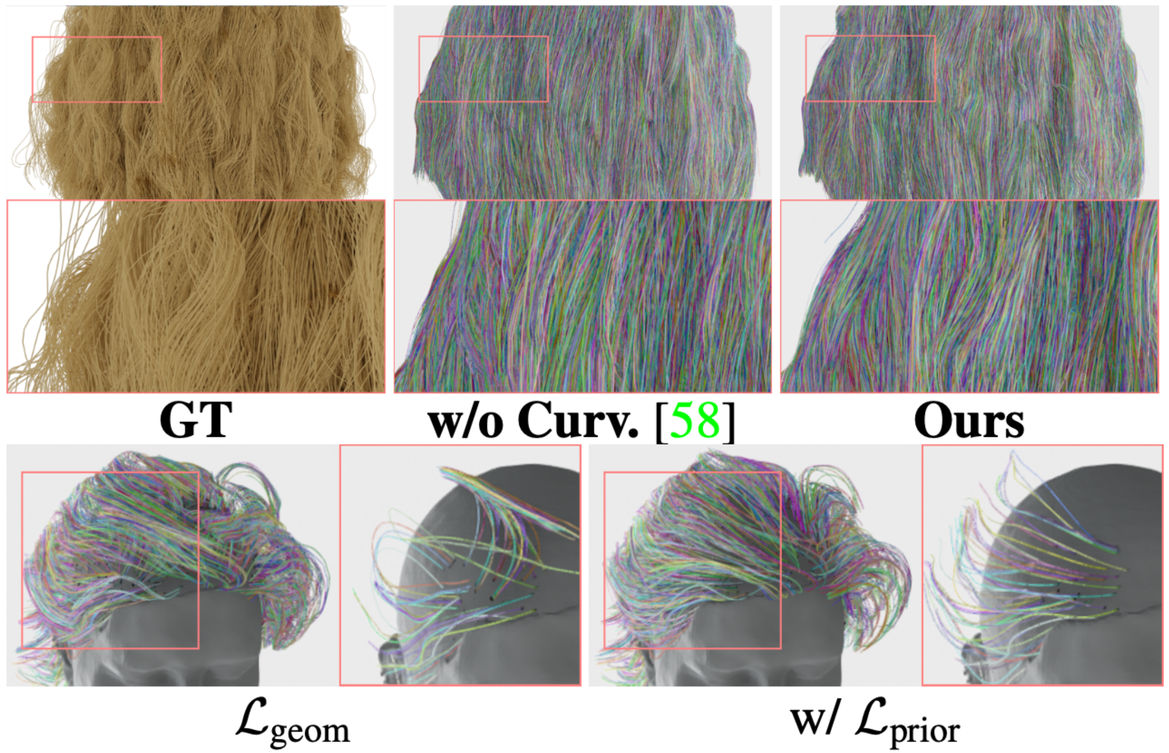


Figure 6: Ablation on curvature (top) and diffusion losses (bottom). The incorporation of curvature loss allows us to better model curly strands, while the diffusion tackles the problems with hair growth directions and unrealistic angles (insets show a subset of hairs for clarity).

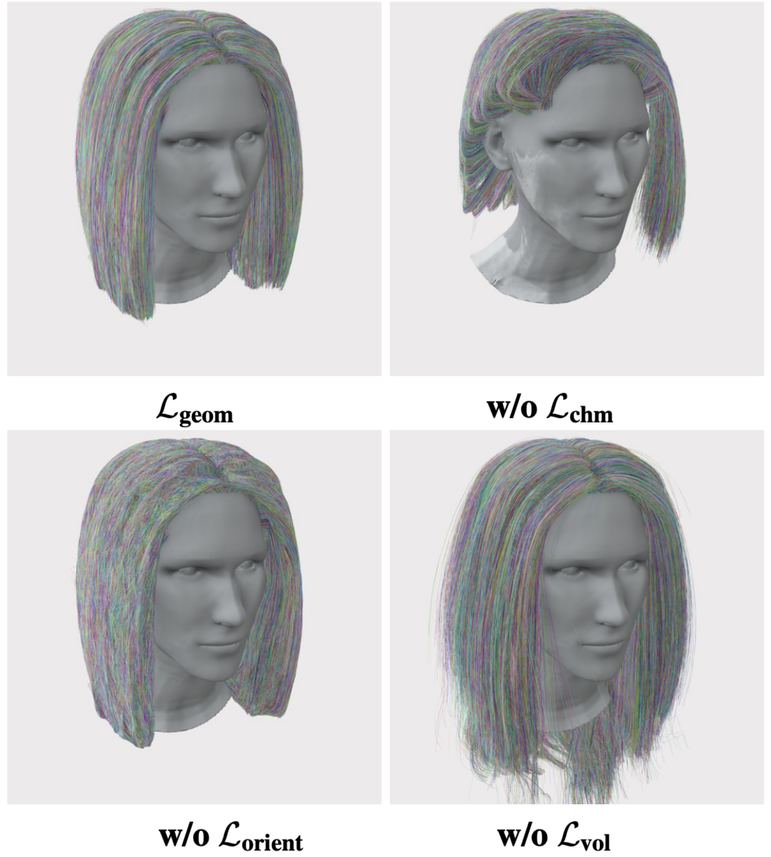


Figure 11: Ablation on individual components of geometry loss $\mathcal{L}_{geom}$. Without chamfer loss $\mathcal{L}_{chm}$ strands doesn't cover the whole hair silhouette. Removing orientation loss $\mathcal{L}_{orient}$ leads to random directions while removing the volume loss $\mathcal{L}_{vol}$ results in uncontrolled strands growing outside the hair region.

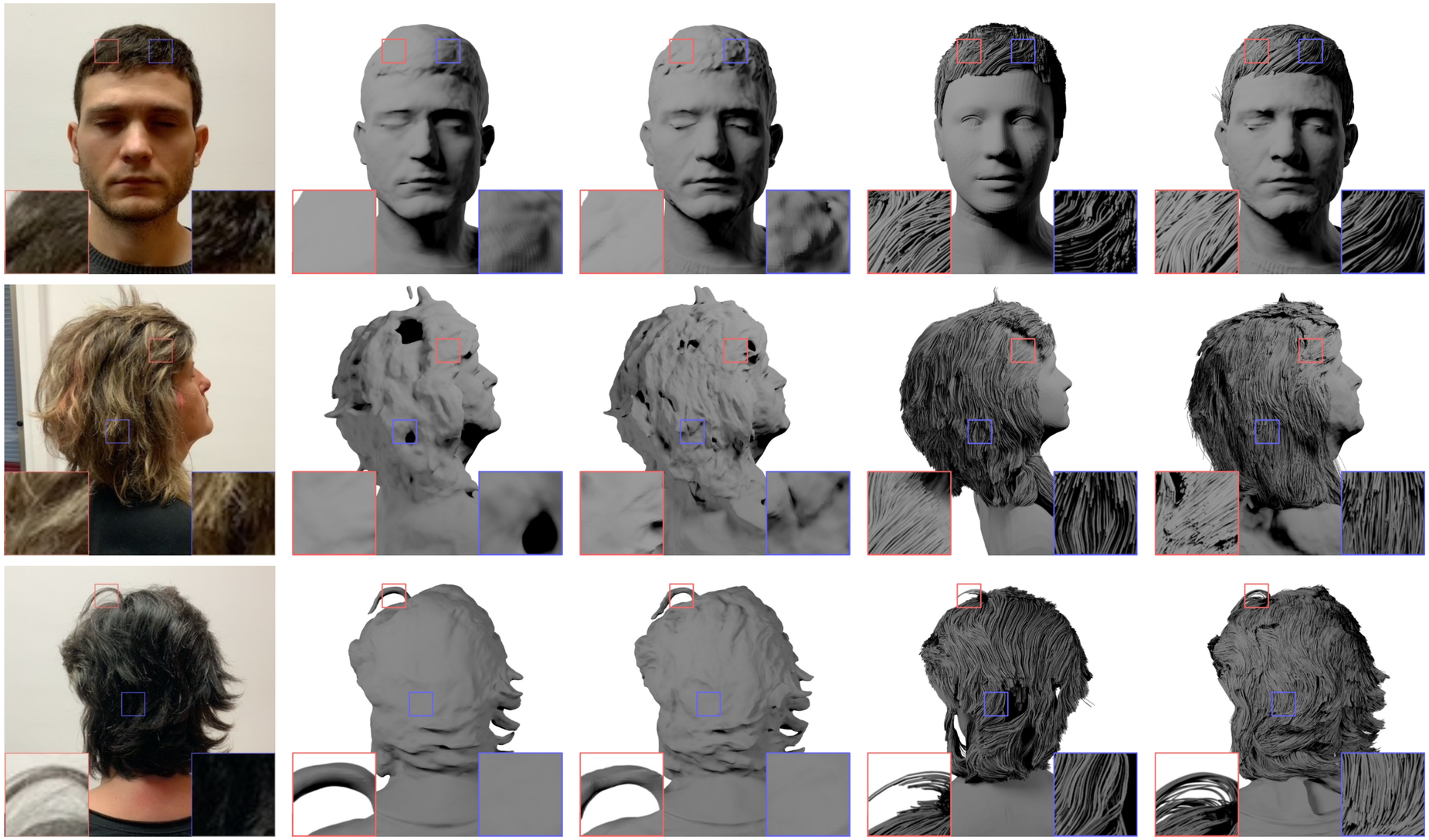


Figure 4: We compare our method with volumetric and strand-based 3D reconstruction systems using a real-world multi-view dataset [56]. While baseline volumetric approaches [46, 66] can only produce coarse hair geometry, our method is able to reconstruct fine details using strands. We also achieve more robust and accurate results than the existing multi-view hair reconstruction methods [30]. For additional results, please refer to the supplementary materials. Digital zoom-in is recommended.

Figure 17: The main limitation of our method is related to curly hair reconstruction, which will be addressed in future work.

Conclusion + recap

- a method for hair modeling that uses **only image- or video-based data without any additional manual annotations**
- employ both **volumetric** and **strand-based** hair representations and combine them with **differential hair rendering** and global **hairstyle priors**
- method can obtain high-fidelity hair reconstructions **even from a monocular video**
- system still **struggles to represent curly hair** and **relies on accurate** hair and body **segmentation masks** to produce the reconstructions