MonoHair: High-Fidelity Hair Modeling from a Monocular Video

Keyu Wu[1] Lingchen Yang[2] Zhiyi Kuang[1] Yao Feng[2,4] Xutao Han[1]
Yuefan Shen[1] Hongbo Fu[3,5] Kun Zhou[1] Youyi Zheng[1†]

[1] Zhejiang University [2] ETH Zurich [3] City University of Hong Kong
[4] Max Planck Institute for Intelligent Systems
[5] Hong Kong University of Science and Technology

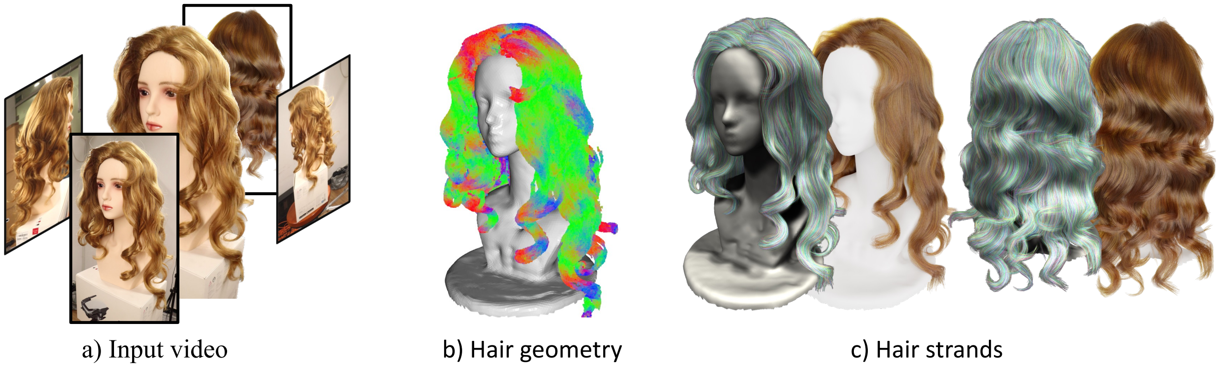

a) Input video
b) Hair geometry
c) Hair strands

Figure 1. We propose a generic framework for 3D hair modeling from monocular videos (a). It commences with a coarse raw geometry produced by volumetric representations. Subsequently, we extract the exterior geometry of the hair from the raw geometry and combine it with an inferred interior structure to obtain the complete 3D hair geometry (b). Finally, we recover the corresponding 3D hair model at the strand level. Our method can reconstruct diverse hairstyles and achieve high-fidelity hair modeling results (c).

Abstract

Undoubtedly, high-fidelity 3D hair is crucial for achieving realism, artistic expression, and immersion in computer graphics. While existing 3D hair modeling methods have achieved impressive performance, the challenge of achieving high-quality hair reconstruction persists: they either require strict capture conditions, making practical applications difficult, or heavily rely on learned prior data, obscuring fine-grained details in images. To address these challenges, we propose MonoHair,a generic framework to achieve high-fidelity hair reconstruction from a monocular video, without specific requirements for environments. Our approach bifurcates the hair modeling process into two main stages: precise exterior reconstruction and interior structure inference. The exterior is meticulously crafted using our Patch-based Multi-View Optimization (PMVO). This method strategically collects and integrates hair information from multiple views, independent of prior data, to produce a high-fidelity exterior 3D line map. This map not only captures intricate details but also facilitates the inference of the hair's inner structure. For the interior, we employ a data-driven, multi-view 3D hair reconstruction method. This method utilizes 2D structural renderings derived from the reconstructed exterior, mirroring the synthetic 2D inputs used during training. This alignment effectively bridges the domain gap between our training data and real-world data, thereby enhancing the accuracy and reliability of our interior structure inference. Lastly, we generate a strand model and resolve the directional ambiguity by our hair growth algorithm. Our experiments demonstrate that our method exhibits robustness across diverse hairstyles and achieves state-of-the-art performance. For more results, please refer to our project page `https://keyuwu-cs.github.io/MonoHair/`

1. Introduction

Hair is a key feature in digital humans, and a detailed 3D hair model will undoubtedly enhance their realism [3, 8, 11, 14, 17]. However, hair modeling is an intricate endeavor, fraught with challenges at every turn. The high

[†] Corresponding author: Youyi Zheng.

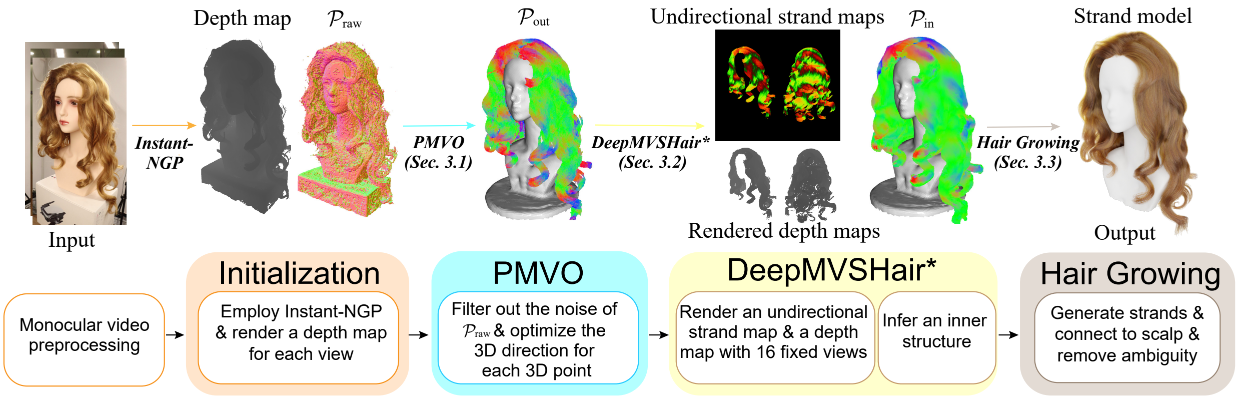

Depth map
$\mathcal{P}_{\text{raw}}$
$\mathcal{P}_{\text{out}}$
Undirectional strand maps
$\mathcal{P}_{\text{in}}$
Strand model
Input
Instant-NGP
PMVO (Sec. 3.1)
DeepMVSHair* (Sec. 3.2)
Hair Growing (Sec. 3.3)
Rendered depth maps
Output
Monocular video preprocessing
Initialization
Employ Instant-NGP & render a depth map for each view
PMVO
Filter out the noise of $\mathcal{P}_{\text{raw}}$ & optimize the 3D direction for each 3D point
DeepMVSHair*
Render an undirectional strand map & a depth map with 16 fixed views
Infer an inner structure
Hair Growing
Generate strands & connect to scalp & remove ambiguity

Figure 2. An overview of our 3D hair reconstruction pipeline.

complexity stems from the unique geometry of individual strands twisting and turning in myriad ways.

In computer graphics, 3D hair is commonly modeled as 3D strands, facilitating rendering and simulation processes. Although accurate strand-based hair modeling can be realized using a light stage [22], it relies on dense synchronized cameras. Besides, the reconstructed hair is incomplete since only the hair's exterior is captured neglecting its inner structure. To address these issues, several multi-view techniques [13, 43] infer the inner structure based on an existing dataset [10] while streamlining the experimental setup, for instance, by using sparse calibrated cameras. However, this sparsity in turn compromises the hair reconstruction quality. Recently, [30] utilizes dense views extracted from a casually-filmed video, achieving better quality and user-friendliness. However, they rely heavily on the data prior and tend to over-smooth the reconstruction results for certain specific hairstyles. This issue is particularly noticeable in hairstyles not well represented in the dataset, such as curly hair types.

Similar issues exist with the data-prior-based multi-view methods. First, data priors, built by learning 3D hair generators with synthetic data, face diminished effectiveness when applied to real data, mainly due to the significant domain gap between the synthetic training data and real-world testing data. Second, the significant reliance on data priors tends to overshadow the rich information contained in the original images. This neglect results in the prior over-dominance, preventing these methods from modeling fine-grained curly hair geometry, typically absent in the current dataset[13, 30, 42, 43]. While such priors are essential for inferring plausible interior hair structures not readily discernible from input data, we believe that the hairstyle's outer layer can be modeled directly and used to enhance and refine the deduced interior structure.

As shown in Fig. 1, to address these concerns, we propose a generic solution for reconstructing hair from a monocular video. Our approach begins by initializing a coarse geometry through learning a Neural Radiance Field (NeRF), followed by sampling around the coarse geometry to generate a dense raw point cloud capable of representing diverse and complex hairstyles. To refine this raw point cloud, we introduce *PMVO*, which leverages the rich information contained in the input video frames to reconstruct a high-quality hair exterior. This approach significantly alleviates the issue of prior over-dominance by focusing on the hair's exterior details. Lastly, to infer the missing inner hair structure, we adopt a data-driven, multi-view hair method, which takes 2D hair information as input. However, instead of applying Gabor filter on the images to get the input, we directly utilize 2D structural renderings derived from the reconstructed exterior, which mirrors the synthetic 2D inputs used during training. This alignment effectively bridges the domain gap between our training data and real-world testing data, thereby enhancing the accuracy and reliability of our interior structure inference.

In summary, the main contributions of our work include:

- We propose a lightweight and generic framework for 3D hair reconstruction from a monocular video. This framework can robustly reconstruct diverse hairstyles, including curly hair, and outperforms state-of-the-art monocular video based hair reconstruction method in reconstruction quality. It also offers a speed improvement of more than tenfold.
- We present a novel process for extracting high-quality exterior hair structures from noisy coarse geometries. This is achieved through our innovative patch-based multiview optimization, incorporating two novel cost functions: ray regularization and patch-wise angular loss.
- We introduce an undirectional strand map, generated by rendering the high-quality hair exterior, to bridge the gap between synthetic and real-world data. This enhances the reliability of interior hair structure inference.

2. Related work

Implicit Representations in 3D Hair Modeling. Recently, implicit representations such as NeRF [19, 21] and implicit surfaces [32, 40] have emerged and gained extensive use in novel view synthesis [1, 2, 19, 37] and general scene reconstruction [7, 23, 33]. Their primary advantages are high-quality results without meticulous camera calibration, and their flexibility for modeling various structures, including hair. For instance, studies like [6, 26, 34–36] employ volumetric representations to implicitly capture hair from multi-view images or monocular videos. However, the hair models generated by these methods fall short of meeting the standards for high-quality 3D strand-based hair reconstruction. Despite this limitation, such representations are valuable for providing initial coarse geometry.

Optimization Based Hair Reconstruction. In the early stages of hair reconstruction research, strand-based representations were the primary focus, beginning with the pioneering work by Paris et al. [24]. This approach set the stage for numerous optimization-based hair reconstruction methods. For instance, Luo et al. [16, 17] optimized a hair mesh to incorporate fine-grained details using hair orientations as constraints. Similarly, other studies [9, 18] employed Multi-View Stereo (MVS) techniques to generate point clouds, optimizing shape primitives like strands and ribbons to create complete hair models. Building upon these methods, Nam et al. [22] introduced a line-based PatchMatch MVS approach, capable of reconstructing high-precision hair segments using a dense capture setup with synchronized cameras. However, their method requires extensive multi-view calibrations and specific lighting conditions, posing practical challenges for average users. Moreover, the reconstructed hair models lack interior structures, thus significantly limiting their applicability.

Hair Reconstruction with Data Prior. Since the USC-HairSalon dataset [10] was released, using data priors for 3D hair reconstruction has gained widespread popularity. Studies like [4, 43] have developed data-driven methods that select and modify hairstyles from a database to match them with the geometry seen in images. Further, research illustrated in [27, 38, 39, 42, 44, 45] has shown how single images can be input into neural networks trained with synthetic data to create strand-based hair models. These deep learning methods also work well with a few images from different views, as shown in [13]. However, they struggle to accurately reconstruct hairstyles not present in the database. Additionally, their effectiveness is reduced when applied to real data, primarily due to the significant domain gap between synthetic training data and real-world testing data. Recently, [30] introduced a novel method that improves hair strand reconstruction by integrating various techniques. This method excels in mitigating issues associated with using synthetic data and produces impressive results. However, it heavily relies on data priors and is less effective in reconstructing curly hair. Please refer to Sec. 4.1 for the discussion.

3. Method

Fig. 2 shows the pipeline of MonoHair. Starting with a set of images $\{I\}$, uniformly sampled from a captured monocular video, we employ NeRF [1, 20, 21] for scene reconstruction. This yields a raw point cloud, $\mathcal{P}_{\text{raw}}$, which represents the hair region, albeit in a noisy manner. Given $\mathcal{P}_{\text{raw}}$, we aim to reconstruct an accurate exterior layer $\mathcal{P}_{\text{out}}$ and infer a plausible interior structure $\mathcal{P}_{\text{in}}$. Based on these two components, we then generate hair strands.

Our PMVO (denoted as Ψ) steps in to carve out a clear hair exterior structure. By leveraging the information from images $\{I\}$, it refines $\mathcal{P}_{\text{raw}}$ and associates each point around the hair's boundary with a 3D hair-growing direction. This process results in a 3D line map, $\mathcal{P}_{\text{out}} = \Psi(\mathcal{P}_{\text{raw}}, I)$, where each 3D line can be represented by its 3D position $\boldsymbol{p}$ and 3D direction $\boldsymbol{d}$: $L^{\boldsymbol{p}} = \{\boldsymbol{p}, \boldsymbol{d}\}$ (Sec. 3.1). Notably, $\mathcal{P}_{\text{out}}$ provides a structured representation of the hair's outer region, enabling the generation of coherent and high-quality 2D hair structure renderings.

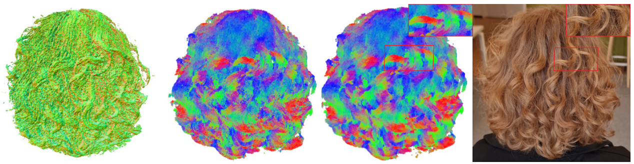
Subsequently, we employ a hair generation network DeepMVSHair [13] and improved it to adjust to our pipeline, denoted as DeepMVSHair* ($\mathcal{N}$). This network is pre-trained on a synthetic 3D hair database and takes in the 2D undirectional strand maps derived from $\mathcal{P}_{\text{out}}$ to deduce the hair's intricate inner structure, represented as $\mathcal{P}_{\text{in}} = \mathcal{N}(\mathcal{R}(\mathcal{P}_{\text{out}}))$ (Sec. 3.2), where $\mathcal{R}$ denotes the rendering operation.

Finally, our strand generation module ζ extracts the hair strands S from the reconstructed outer layer $\mathcal{P}_{\text{out}}$ and the inferred inner structure $\mathcal{P}_{\text{in}}$ (Sec. 3.3). This can be formulated as:

$$S = \zeta(\mathcal{P}_{\text{out}}, \mathcal{P}_{\text{in}}). \tag{1}$$

3.1. Patch-based Multi-View Optimization

As discussed in Sec. 1, a fine-grained exterior structure is essential for hair reconstruction. However, obtaining such a high-precision exterior hair structure is nontrivial. Inspired by [22], integrating multi-view image information to produce a 3D line map $\mathcal{P}_{\text{out}}$ is a possible solution. However, their method reconstructs the exterior from scratch with dense calibrated images and is highly limited by expensive capture equipment and strict capture conditions. In response to these limitations, Neural Haircut [30] employs NeuS [32] to first initialize a coarse 3D hair geometry and then refines it through differentiable rendering and a learned data prior. However, this approach applies the data prior to the overall hair shape, causing a loss of fine-grained hair details. Additionally, the representation of hair geometry using a signed distance field (SDF) further exacerbates the



(a) Raw point cloud (b) Line map (c) Refined line map (d) Image

Figure 3. Visualization of the line map extracted from raw geometry.

over-smoothness. Thus, our method reconstructs the exterior geometry of hair without any hair data prior constraint, preserving as many details from the image as possible.

Specifically, we first initialize the coarse hair structure using the point cloud extracted from NeRF [1, 20, 21] instead of NeuS [32]. For efficiency, we utilize the Instant-NGP framework [21] and obtain an initial coarse geometry by applying a threshold of 2.5 to the learned density values. Subsequently, we densely sample around the obtained coarse geometry to obtain $\mathcal{P}_{\text{raw}}$. Our key observation is that although $\mathcal{P}_{\text{raw}}$ is very noisy and even terrible, it encompasses nearly all of the hair's exterior geometry, as illustrated in Fig. 3 (a). Thus, our proposed PMVO is designed to first eliminate the noise (points not belonging to the hair) while preserving the fine-grained geometry of the hair and then calculate a 3D growing direction at each remaining point.

Input Data. The input of our PMVO consists of the coarse point cloud $\mathcal{P}_{\text{raw}}$ and an image set, uniformly sampled from the captured monocular video. Subsequently, for each image, we obtain three maps: a 2D orientation map $\mathcal{O}$, a confidence map $\mathcal{C}$, and a depth map $\mathcal{D}$. The 2D orientation map $\mathcal{O}$ and the confidence map $\mathcal{C}$ are extracted using the Gabor filter similar to the previous work [24]. $\mathcal{O}$ contains the 2D hair growth direction at each pixel while $\mathcal{C}$ measures the confidence of the estimated direction. The depth map $\mathcal{D}$ is obtained by directly rendering the depth of $\mathcal{P}_{\text{raw}}$ with the estimated camera parameters from COLMAP [28]. $\mathcal{D}$ serves as the cue for judging the visibility $V^{\boldsymbol{p}}$ of point $\boldsymbol{p} \in \mathcal{P}_{\text{raw}}$ by: $V^{\boldsymbol{p}} = 1 - \frac{\boldsymbol{p}_z - \mathcal{D}(\Pi(\boldsymbol{p}))}{\tau}$, where $\boldsymbol{p}_z$ is the z coordinate of $\boldsymbol{p}$ and Π is the projection function. τ represents a visible threshold, which we set to 5mm in our experiment to differentiate between the exterior (visible regions) and inner regions (invisible regions) of the hair. These maps will be used to calculate our cost function, as described below.

Cost Function. For each point $\boldsymbol{p} \in \mathcal{P}_{\text{raw}}$, our PMVO attempts to find a correct 3D line $L^{\boldsymbol{p}}$ by integrating information from all views in which $\boldsymbol{p}$ is visible. Here, the correct 3D line should minimize our proposed cost function $\mathcal{L}_{opt}$, which describes the similarity between the projected 2D line $l^{\boldsymbol{p}}$ of $L^{\boldsymbol{p}}$ and the 2D orientation in the corresponding views as follows:

$$\mathcal{L}_{opt}(\boldsymbol{p}, L^{\boldsymbol{p}}) = \frac{\sum_{i=1}^{N} w_i g_i(\mathcal{O}_i(\Pi_i(\boldsymbol{p})), l_i^{\boldsymbol{p}})}{\sum_{i=1}^{N} w_i}, \tag{2}$$

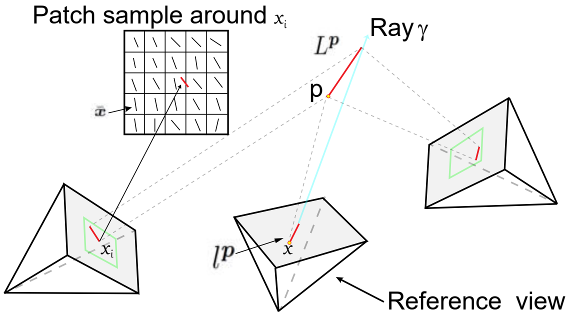


Figure 4. Schematic diagram of patch-based multi-view optimization.

where i denotes a specific view, $l_i^{\boldsymbol{p}} = \Pi_i(L^{\boldsymbol{p}})$, N is the number of views used, and $w_i = V_i^{\boldsymbol{p}} \cdot \mathcal{C}_i(\Pi_i(\boldsymbol{p}))$ is the weight of $\boldsymbol{p}$ in the ith view. $g_i(\mathcal{O}_i(\Pi(\boldsymbol{p})), l_i^{\boldsymbol{p}})$ is our proposed patch-wise angular loss function. This function measures the angle difference between the 2D orientation $\mathcal{O}_i(\Pi_i(\boldsymbol{p}))$ corresponding to point $\boldsymbol{p}$ and the projected 2D line direction $l_i^{\boldsymbol{p}}$ in the ith frame. The key idea for extracting the hair line map $\mathcal{P}_{\text{out}}$ from the raw point cloud $\mathcal{P}_{\text{raw}}$ is that each 3D line $L^{\boldsymbol{p}}$ belonging to the hair can be projected into all visible views, and its 3D direction projections align with the corresponding 2D orientations in those views—a characteristic that noise does not possess. Thus, our objective is to find the best-fitting $L^{\boldsymbol{p}}$ that minimizes the cost function $\mathcal{L}_{opt}$:

$$L^{\boldsymbol{p}} = \arg\min_{L^{\boldsymbol{p}}} \mathcal{L}_{opt}(\boldsymbol{p}, L^{\boldsymbol{p}}). \tag{3}$$

Patch-wise Angular Loss. Our patch-wise angular loss aims to measure the re-projection cosine difference between the projected 2D line and the corresponding 2D orientation. As shown in Fig. 4, for the 2D projection $\boldsymbol{x} = \Pi(p)$ of $\boldsymbol{p}$ at each frame, we sample k number of 2D points centered on $\boldsymbol{x}$, $k = r^2$, where r is a patch size (set as 5 pixels in our experiments). Then, we can formulate our patch-wise angular loss g_i as:

$$g_i(\mathcal{O}_i(\boldsymbol{x}), l_i^{\boldsymbol{p}}) = \sum_{\bar{\boldsymbol{x}} \in X_i(\boldsymbol{x})} \mathcal{C}_i(\bar{\boldsymbol{x}}) \cdot (1 - cos(\mathcal{O}_i(\bar{\boldsymbol{x}}), l_i^{\boldsymbol{p}})), \tag{4}$$

where X_i is a 2D point set sampled 2D centered on $\boldsymbol{x}$ in the ith view. This loss function has two advantages. First, it allows some calibration errors. Second, it can also smooth the direction of locally adjacent 3D lines, the same as the properties of hair.

Optimization. As shown in Fig. 4, to solve the cost function $\mathcal{L}_{opt}$ robustly, we first select a reference frame with the largest confidence in the frames where $\boldsymbol{p}$ is visible to initialize $L^{\boldsymbol{p}}$ with a 3D vector $L^{\boldsymbol{p}} = (\mathcal{O}(\Pi(\boldsymbol{p})), 0)$. It's easy to find that the correct 3D line $L^{\boldsymbol{p}}$ should intersect with the ray γ emitted from the other end of the 2D line $l^{\boldsymbol{p}}$ at the reference view. Then, in our process of optimizing $L^{\boldsymbol{p}}$, we constrain the direction of $L^{\boldsymbol{p}}$ by the following regulariza-

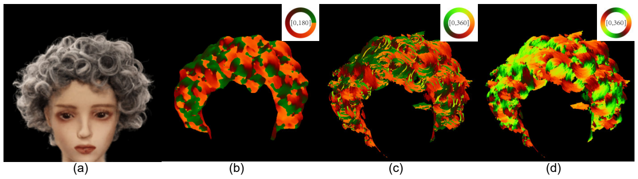


Figure 5. Visualization of different 2D hair growth direction maps. (a) Portrait image. (b) Orientation map reported in [13]. (c) Rendered strand map. (d) Rendered undirectional strand map.

tion:

$$\mathcal{L}_{reg} = dist(\gamma, Ray(\boldsymbol{p}, L^{\boldsymbol{p}})), \tag{5}$$

where $Ray(\boldsymbol{p}, L^{\boldsymbol{p}})$ represents the ray starting from point $\boldsymbol{p}$ along the $L^{\boldsymbol{p}}$ direction, and $dist(\cdot)$ is the function for calculating the closest distance between two rays.

Line Refinement. For each $\boldsymbol{p} \in \mathcal{P}_{\text{raw}}$, we filter out the inner points, which are not visible in all frames. Then we keep the point $\boldsymbol{p}$ with $\mathcal{L}_{opt}(\boldsymbol{p}, L^{\boldsymbol{p}}) < 0.05$ (about 15 degrees) as the final $\mathcal{P}_{\text{out}}$, as shown in Fig. 3 (b). However, the extracted hair's exterior geometry suffers from some noise. Therefore, we find 100 neighbors of $\boldsymbol{p}$, denoted as $\{\boldsymbol{p}_{nei}\}$, and calculate the variance between $\boldsymbol{p}$ and $\{\boldsymbol{p}_{nei}\}$:

$$var(\boldsymbol{p}) = cos(L^{\boldsymbol{p}}, \boldsymbol{avg}(\mathcal{L}^{\boldsymbol{p}_{nei}})). \tag{6}$$

Subsequently, we update the $L^{\boldsymbol{p}}$ with $\boldsymbol{avg}(\mathcal{L}^{P_{nei}})$ if $var(\boldsymbol{p}) > 0.015$ (about 10 degrees) to produce the final exterior of hair $\mathcal{P}_{\text{out}}$, as shown in Fig. 3 (c).

3.2. Infer Interior Geometry

For a complete acquisition of hair geometry, inferring the hair's inner structure is necessary. We employ a method that incorporates data priors, similar to DeepMVSHair[13]. Their method takes multi-view calibrated images as input and trains a HairMVSNet on a synthetic dataset to integrate multi-view hair structure features to infer the hair geometry, represented as a pair of a 3D occupancy field and a 3D orientation field. However, the direct application of this approach to our problem faces two challenges: 1) the need for calibrated images and 2) a domain gap between synthetic and real 2D data. The 2D orientation maps, extracted using Gabor filters from images, are limited by capture quality and thus introduce directional ambiguity, a common issue in 3D hair modeling.

To overcome these limitations, we propose two improvements : 1) Instead of extracting 2D orientation maps from calibrated images, we render the extracted exterior layer of hair, $\mathcal{P}_{\text{out}}$, to 16 fixed synthetic views (**DeepMVSHair*** is trained using these fixed 16 views), as shown in Fig. 5. Our key observation is that independently extracting its own 2D orientation map from each image is easily affected by image quality, viewing angle, and occlusion. In contrast, integrating geometric information from multi-view images into 3D and rendering it into 2D results in clearer geometry and greater robustness. Furthermore, this process is the same as the training data preparation. Additionally, this strategy facilitates the rendering of additional views, enabling us to produce images from numerous angles and positions while ensuring the precision of camera parameter settings. 2) While [44] attempted to train a neural network to mitigate the ambiguity only in the frontal view, it is insufficiently robust for other views. Our observations suggest that resolving ambiguities in the 3D space is often simpler than directly in the image space. Therefore, we tackle this issue in a subsequent stage (Sec. 3.3) by defining a 2D undirectional strand map $\mathcal{U}$ as:

$$\mathcal{U} = (\cos(2 \cdot \mathcal{O}), \sin(2 \cdot \mathcal{O})), \tag{7}$$

where $\mathcal{O} \in [0, 180]$. Here, $\mathcal{O}$ and $\mathcal{O} + 180$ are encoded into the same color space to better facilitate the training of DeepMVSHair [13]. Concurrently, we render a depth map $\mathcal{D}$ for each view using our high-quality exterior hair structure. Consequently, we can infer the inner structure using the improved **DeepMVSHair*** as follows:

$$\mathcal{P}_{\text{in}} = \mathcal{N}(\mathcal{P}_{\text{out}}, \mathcal{U}, \mathcal{D}). \tag{8}$$

Besides, to accommodate the aforementioned modifications, we also design a new loss function for their orientation prediction component. Specifically, the predicted 3D direction d and its opposite direction $-d$ are considered to be the same direction. This can be formulated by:

$$\mathcal{L}_{ori} = \frac{1}{N}\sum_{i}^{N} \min(\frac{||\hat{d} - d||_1}{3}, \frac{||\hat{d} + d||_1}{3}), \tag{9}$$

where N is the number of views, $\hat{d}$ is the ground truth. For more details, please refer to [13].

3.3. Strand Generation

To generate a complete 3D strand model, we need to merge $\mathcal{P}_{\text{out}}$ and $\mathcal{P}_{\text{in}}$. Specifically, for each point $\boldsymbol{p}$ in $\mathcal{P}_{\text{in}}$, we selectively integrate the points that are invisible in all views, combining them with $\mathcal{P}_{\text{out}}$ to form our final hair geometry $\mathcal{H}$. This strategy ensures that the data prior does not overshadow the details in the hair's exterior geometry. Subsequently, we voxelize the space and convert $\mathcal{H}$ to a high-resolution 3D orientation field and then use forward Euler and backward Euler to generate segments $\{s\}$ similar to previous works [13, 38]. The difference is that we recursively connect short segments into long strands instead of connecting to the scalp root directly since it is difficult to distinguish which end of a segment is the root when the short segments are close to the scalp. Besides, we connect long strands to the scalp and detect the direction of unconnected strands using connected strands to resolve direction ambiguity. Algorithm 1 presents our growing step in detail.

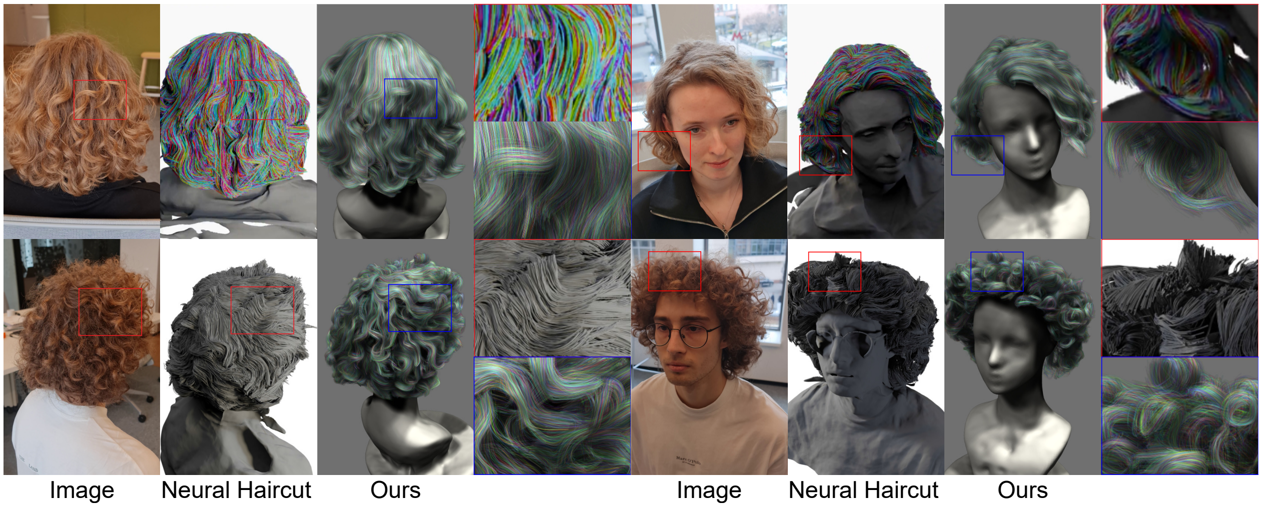


Figure 6. Qualitative comparison with Neural Haircut [30], which has limited ability to reconstruct complex hairstyles, especially for curly hair (the reconstruction results are all from their paper). On the contrary, our reconstruction results can maintain more details in the images.

Algorithm 1: Strand Generation

```
Input: Hair scalp $\Omega$, hair segment set $\{s\}$
Output: Strands connected to scalp $S^c$
Step1: For each segment $s$, find the nearest
  neighbor segment to root ($s^r_{nei}$) and tip ($s^t_{nei}$)
  respectively.
Step2: Recursively connect ($s^r_{nei}$) and ($s^t_{nei}$) to
  produce long strand set $\{s^l\}$
while $len(\{s^l\}) \neq 0$ do
    Step3: For each $s^l$, calculate the distance
      $dist(s^l, \Omega)$ of its end (either root or tip) closest
      to the scalp. If $dist(s^l, \Omega) < 15mm$, include $s^l$
      to the set $S^c$ and remove it in $\{s^l\}$.
    step4: For each $s^l$, find the neighbor $s^c$ in $S^c$
      with a distance $dist(s^l, s^c) < 5mm$, having
      the most similar growth direction, then resolve
      the ambiguity in growth direction based on it.
      Where $s^c \in S^c$.
    Step5: For the remaining $s^l$, find the neighbor
      $s^c$ in $S^c$ with a distance $dist(s^l, s^c) < 2mm$,
      and connect it with the closest point on $s^c$.
end
```

Method	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40
	Precision			**Recall**			**F-score**		
DeepMVSHair [13]	43.9	67.2	79.5	9.2	19.5	24.8	15.2	30.2	37.8
DeepMVSHair*	49.5	77.1	86.3	9.3	19.0	25.1	15.7	30.5	38.9
Neural Haircut [30]	52.9	78.1	88.4	9.8	17.8	**26.3**	16.4	28.7	40.3
Ours	**60.8**	**83.3**	**92.1**	**10.4**	**19.3**	25.9	**17.8**	**31.3**	**40.4**

Thresholds: mm / degrees

Table 1. Quantitative comparison with [13, 30]. Our method achieves the highest precision and F-score.

Method	coarse geometry	fine geometry	strand generate	total
Neural Haircut [30]	24-36h	48-72h	/	72-108h
Ours	5min	3-4h	1-2h	4-6h

Table 2. Comparison with Neural Haircut [30] in terms of time consumption. Our method is ten times faster.

4. Evaluation

We train our improved DeepMVSHair* model on USC-HairSalon [10], which includes 343 hairstyles aligned with a template head. To augment the dataset, we applied random translations, rotations, and scaling, resulting in a total of 2,744 strand models. We evaluate our method through a comprehensive evaluation, encompassing both quantitative and qualitative comparisons, using synthetic data [41] and public real-world H3DS dataset [25] as well as real-world monocular video captures (Sec. 4.1). We also conduct an ablation study to evaluate the importance of each component of our method (Sec. 4.2). Implementation details and more experiments please refer to supplementary materials.

4.1. Comparison

Baselines. We compare MonoHairwith strand-based hair modeling methods [13, 30, 38, 44], as well as a popular 3D reconstruction method [32] and a NeRF-based method [21]. Where **Neural Haircut** [30] reconstructs a strand model from a monocular video, consistent with our input. **DeepMVSHair** [13] is a strand-based reconstruction method based on sparse multi-view images. In our evaluation, we compare our method with basic **DeepMVSHair** and our improved implementation (**DeepMVSHair***), using both synthetic [41] and real-world data. **NeuS** [32] is a representative reconstruction method based on an SDF representation, while **Instant-NGP** [21] is commonly used for novel view synthesis. We compare our method with these methods and

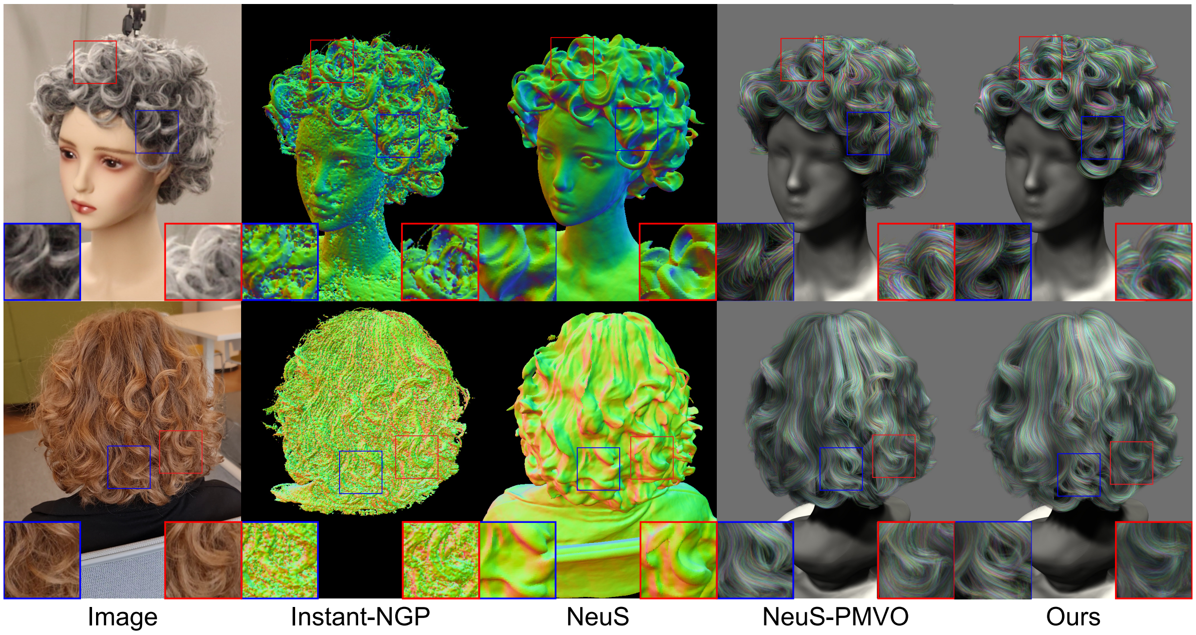


Figure 7. Qualitative comparison with Instant-NGP [21] and NeuS [32]. These volumetric approaches can only produce coarse hair geometry. We also compare the results with another initialization method (NeuS-PMVO). Although they yield similar results, initializing coarse geometry using NeuS tends to obscure fine-grained details.

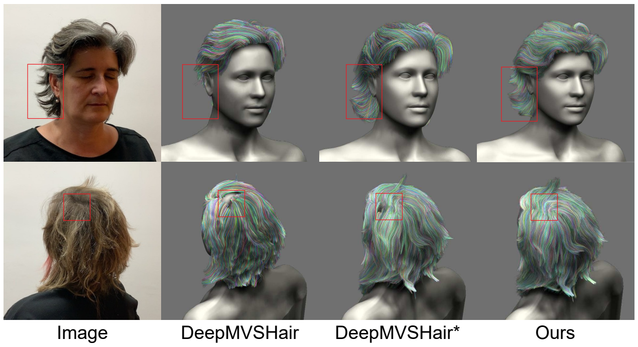


Figure 8. Qualitative comparison with DeepMVSHair [13] on the H3DS[25] dataset. Since the 2D orientation maps extracted from the images may be inconsistent across different views, their results cause some lost geometry. Our improved DeepMVSHair* effectively addresses this issue, though there are still some details lost due to data prior limitations. In contrast, our method escapes these limitations and achieves a high-fidelity result with richer details.

NeuS-PMVO, where **NeuS-PMVO** is our method with the SDF representation as coarse initialization geometry. Finally, we also provide comparisons with single-view based methods **NeuralHDHair**[38] and **HairStep** [44] in the supplementary materials.

Qualitative comparison with **Neural Haircut** [30] is shown in Fig. 6. Their approach tends to yield straight strands and lacks the ability to effectively represent curly hair, primarily due to the imposition of overly strong constraints by the data prior. Besides, the hair geometry representation using SDF also limits some curly strands into a smooth surface (see below for more discussion). In contrast, it is evident that our method exhibits greater robustness for curly hair. We also conduct a quantitative comparison with them [30] on the synthetic dataset [41]. The comparison results are shown in Tab. 1. Our method achieves the highest precision and F-score. Moreover, we also compare the two methods in terms of reconstruction efficiency as shown in Tab.2. **Neural Haircut** takes 3-4 days for each subject on a single NVIDIA RTX 3090, which significantly limits its practical application, while ours only takes 4-6 hours.

We provide qualitative comparisons with **Instant-NGP**, **NeuS**, and **NeuS-PMVO**. **Instant-NGP** and **NeuS** can only produce a coarse hair geometry, our method can achieve more robust and accurate results than them. It is important to note that the results obtained by NeRF-based methods often exhibit lots of noise, and **NeuS** is more effective in obtaining clean 3D geometry. However, as shown in the last two columns of Fig. 7, the results of **NeuS-PMVO** are smoother in some details, and thus we choose **Instant-NGP** to initialize the coarse geometry.

Quantitative and qualitative comparisons with **DeepMVSHair** are given in Tab. 1 and Fig. 8, respectively. While their method is capable of generating plausible geometry, its performance is significantly impacted by the quality of the input 2D orientation map. On the other hand, taking in undirectional strand maps derived from the 3D line map, the proposed **DeepMVSHair*** can produce a better result. However, since the learning method is limited by the distribution of the data prior, the obtained results may differ

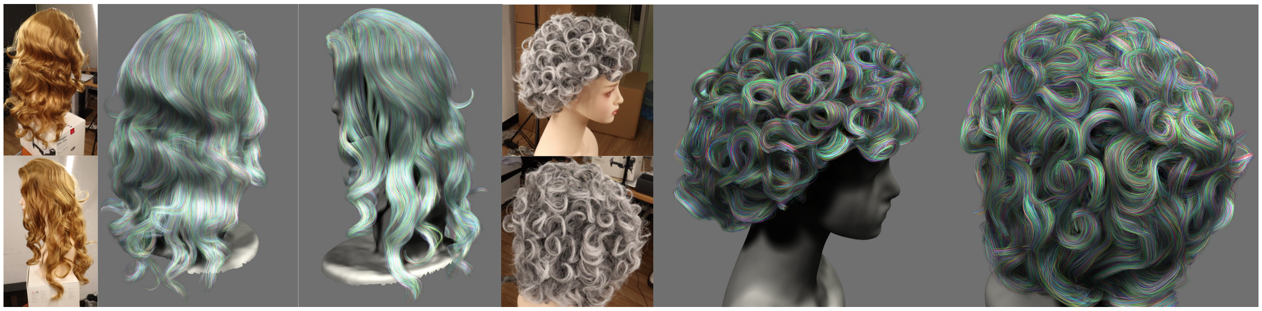


Figure 9. Given a monocular video, our method can reconstruct a high-fidelity strand model, including intricate curly hair. For more results please refer to the supplementary materials.

from the captured images in detail. In contrast, our method synthesizes a high-quality exterior structure of the hair and only applies data priors to the invisible inner geometry, resulting in superior results.

As shown in Fig. 9, we also provide some challenging cases of real-world video capture to evaluate our method. Our method can robustly reconstruct diverse hairstyles, including straight, wavy, and curly hair, and achieve high-quality and realistic results. For more examples, please refer to our supplementary materials.

4.2. Ablation Study

We evaluate the performance of each component of our method via an ablation study on real data and synthetic data [41]. As shown in Fig. 10, without PMVO, errors in camera parameters lead to the removal of many hair points as noise, resulting in the loss of some hair strands. On the other hand, when DeepMVSHair* is not applied, the results lack internal structures, appearing as a shell comprised of isolated segments. For more ablation studies please refer to our supplementary materials.

5. Conclusion and Discussion

In this paper, we have rethought the existing multi-view based hair reconstruction pipeline, where most methods apply the learned data prior directly to the reconstruction of the entire hairstyle, which is extremely limited by the diversity of training databases. To this end, we proposed MonoHair, a generic framework that bifurcates hair modeling into exterior and interior geometries. It extracts the exterior hair structure from multiview images without relying on data priors. Subsequently, the framework deduces the inner hair structure by combining learned data priors with the extracted high-quality exterior hair structure. Extensive experiments demonstrated that our method employing coarse geometry produced by [21] combined with the proposed PMVO and inner inference module can can reconstruct a high-fidelity strand model and support various hairstyles, including curly hair.

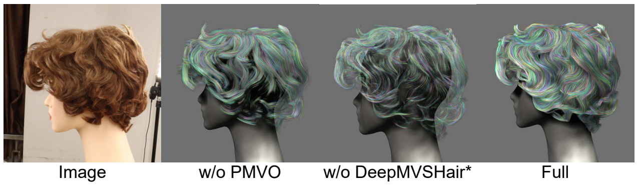


Figure 10. Qualitative evaluation of each key component of our method. PMVO helps produce a high-quality exterior hair geometry. While DeepMVSHair* helps infer the internal geometry to obtain a complete hair geometry.

As shown in Fig. 9 and Fig. 11, the main limitation of our method is that, while we can successfully reconstruct the majority of hair geometry, due to severe intersections and occlusions, some intricate hairstyles (such as braids), the connection relationships may be incorrect. This is primarily because despite we can reconstruct high-quality hair exterior, the inner geometry remains dependent on the data prior. In principle, expanding the dataset to include a more diverse range of hairstyles or directly reconstructing the interior structure using computed tomography similar to CT2Hair [29] can alleviate this limitation.

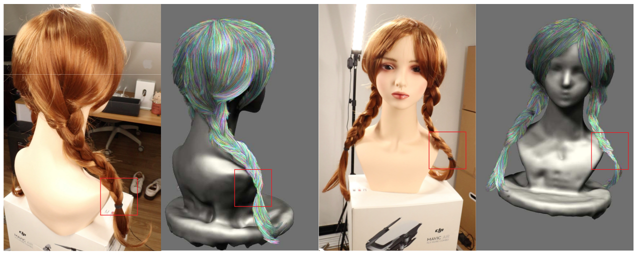


Figure 11. The main limitation of our method is that the connection relationships may be incorrect in instances with severe intersections and occlusions.

Acknowledgements

This work is supported in part by the NSF China (No. 62172363). Besides, We thank the State Key Lab of CAD&CG for providing the computational resources for this work. We also sincerely thank Vanessa Sklyarova for providing the data and reconstruction results to compare with Neural Haircut.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3, 4

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3

[3] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International journal of computer vision*, 107(2):177–190, 2014. 1

[4] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Transactions on Graphics*, 35(4), 2016. 3

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2023. 11, 14

[6] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J. Black. Learning disentangled avatars with hybrid 3d representations. *arXiv*, 2023. 3

[7] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022. 3

[8] Sunil Hadap, Marie-Paule Cani, Ming Lin, Tae-Yong Kim, Florence Bertails, Steve Marschner, Kelly Ward, and Zoran Kačić-Alesić. Strands and hair: modeling, animation, and rendering. In *ACM SIGGRAPH 2007 courses*, pages 1–150. 2007. 1

[9] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics*, 33(4):1–10, 2014. 3

[10] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015. 2, 3, 6, 11

[11] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Transactions on Graphics (ToG)*, 36(6):1–14, 2017. 1

[12] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1140–1147, 2022. 11

[13] Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepmvshair: Deep hair modeling from sparse views. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 2, 3, 5, 6, 7

[14] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. Facial performance sensing head-mounted display. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015. 1

[15] Kunliang Liu, Ouk Choi, Jianming Wang, and Wonjun Hwang. Cdgnet: Class distribution guided network for human parsing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4473–4482, 2022. 11

[16] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *CVPR*, 2012. 3

[17] Linjie Luo, Hao Li, and Szymon Rusinkiewicz. Structure-aware hair capture. *ACM Transactions on Graphics*, 32(4): 1–12, 2013. 1, 3

[18] Linjie Luo, Cha Zhang, Zhengyou Zhang, and Szymon Rusinkiewicz. Wide-baseline hair capture using strand-based refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 265–272, 2013. 3

[19] 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. In *ECCV*, 2020. 2, 3

[20] 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. *Communications of the ACM*, 65(1):99–106, 2021. 3, 4

[21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3, 4, 6, 7, 8, 11

[22] Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 155–164, 2019. 2, 3

[23] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 3

[24] Sylvain Paris, Hector M Briceno, and François X Sillion. Capture of hair geometry from multiple images. *ACM transactions on graphics (TOG)*, 23(3):712–719, 2004. 3, 4

[25] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021. 6, 7

[26] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *European Conference on Computer Vision*, pages 73–89. Springer, 2022. 3

[27] Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 3d hair synthesis using volumetric variational autoencoders. *ACM Transactions on Graphics*, 37(6):1–12, 2018. 3

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 4, 11

[29] Yuefan Shen, Shunsuke Saito, Ziyan Wang, Olivier Maury, Chenglei Wu, Jessica Hodgins, Youyi Zheng, and Giljoo Nam. Ct2hair: High-fidelity 3d hair modeling using computed tomography. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023. 8

[30] Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. Neural haircut: Prior-guided strand-based hair reconstruction. In *ICCV*, 2023. 2, 3, 6, 7, 11, 12

[31] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3667–3676, 2020. 11

[32] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 3, 4, 6, 7

[33] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 3

[34] Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. Learning compositional radiance fields of dynamic human heads. In *CVPR*, 2021. 3

[35] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Hvh: Learning a hybrid neural volumetric representation for dynamic hair performance capture. In *CVPR*, 2022.

[36] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Chen Cao, Jason Saragih, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Neuwigs: A neural dynamic model for volumetric hair capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8641–8651, 2023. 3

[37] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, pages 16210–16220, 2022. 3

[38] Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2022. 3, 5, 6, 7, 11

[39] Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. Dynamic hair modeling from monocular videos using deep neural networks. *ACM Transactions on Graphics (TOG)*, 38 (6):1–12, 2019. 3

[40] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[41] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Transactions on Graphics (TOG)*, 28(5):1–7, 2009. 6, 7, 8, 11, 12

[42] Meng Zhang and Youyi Zheng. Hair-gan: Recovering 3d hair structure from a single image using generative adversarial networks. *Visual Informatics*, 3(2):102–112, 2019. 2, 3

[43] Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. A data-driven approach to four-view image-based hair modeling. *ACM Trans. Graph.*, 36(4):156–1, 2017. 2, 3

[44] 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. In *CVPR*, 2023. 3, 5, 6, 7, 11

[45] Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. Hairnet: Single-view hair reconstruction using convolutional neural networks. In *ECCV*, 2018. 3

MonoHair: High-Fidelity Hair Modeling from a Monocular Video

Supplementary Material

6. Implementation Details and Data Preprocessing

6.1. Datasets preprocessing

Synthetic data. We train the DeepMVSHair* on USC-HairSalon[10], which contains 343 hairstyles aligned with a template head. We augment the dataset to 2,744 samples with three degrees of freedom: scaling, rotation, and translation. Then, for each sample, we generate a strand model, the ground-truth 3D orientation, and a rendered 2D unidirectional map with 16 fixed views. We also evaluate our method on another synthetic dataset [41]. Specifically, for each model, we use Blender [5] to render 150 RGB images with a resolution of 1920×1080, employing different camera poses. Then, we reconstruct the corresponding strand model and evaluate it with the ground truth.

Real-world data. We also evaluate our method utilizing a public multi-view H3DS dataset [41] and a set of real-world captured monocular videos. For the H3DS dataset, each scene has 32 views with their corresponding camera parameters. For monocular videos, to better initialize the coarse geometry, we select 150-200 frames around the subject using an image quality assessment network [31] and employ structure-from-motion with COLMAP [28] to obtain the camera intrinsic and extrinsic parameters. Then, we calculate the 2D orientation map utilizing 180 Gabor filters with $\sigma_x = 1.8$, $\sigma_y = 2.4$, frequency $\omega = 0.25$, and kernel size $k = 17$. Lastly, we also generate hair masks using human matting [12] and semantic segmentation [15].

6.2. Implementation details

To initialize the coarse point cloud, we densely sample around the coarse geometry produced by Instant-NGP [21] at a high resolution of $512 \times 512 \times 384$. For the PMVO, we remove the low confidence area when performing the patch sample strategy. Besides, we select 10 reference frames based on confidence values for different initializations of L^p to calculate $\mathcal{L}_{opt}$ more robustly. For interior hair geometry inference, we grow short segments using $\mathcal{P}_{\text{out}}$ to render the undirectional strand maps. Then, we train DeepMVSHair* on 16 fixed views at 1280×720 resolution with the learning rate of 0.0005. The training process takes about 6 days on a single NVIDIA RTX 3090, 3 days for the 3D orientation prediction network and 3 days for the 3D occupancy prediction network.

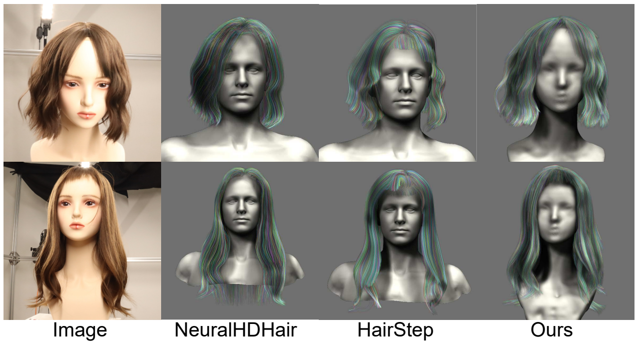


Figure 12. Qualitative comparison with single view-based hair modeling methods. The reconstruction results are produced by the open source code and pre-trained models.

7. Additional Experiments and Results

7.1. More comparisons

Comparison with single view-based image methods. We provide an extended qualitative comparison with the single view-based reconstruction methods [38, 44]. As shown in Fig. 12, obviously, our method achieves higher fidelity results.

Extended comparison with Neural Haircut. We provide an additional comparison with Neural HairCut [30] in Fig. 13. While their method can handle both long and short hair, it struggles to accurately reconstruct some complex hairstyles, such as curly hair. In contrast, our approach is versatile and adaptable to more hairstyles.

7.2. Additional ablation studies

Ablation studies on synthetic data. We also evaluate each component of our method on a synthetic dataset [41]. We render the ground-truth 3D strand model into RGB images using Blender and reconstruct it. As shown in Fig. 14. PMVO helps us maintain high-fidelity exterior hair geometry while DeepMVSHair* assists in reconstructing the complete strand model. As indicated in Tab. 3, without PMVO, both precision and recall are greatly reduced due to the errors of camera parameters. On the other hand, without DeepMVSHair*, we can achieve the highest precision but the lowest recall, since the reconstruction results contain only the outer layer of hair. Finally, our full method combining PMVO and DeepMVSHair* achieves the highest recall and F-score.

Evaluation of DeepMVSHair*. To evaluate the effectiveness of our improvements to DeepMVSHair, we con-

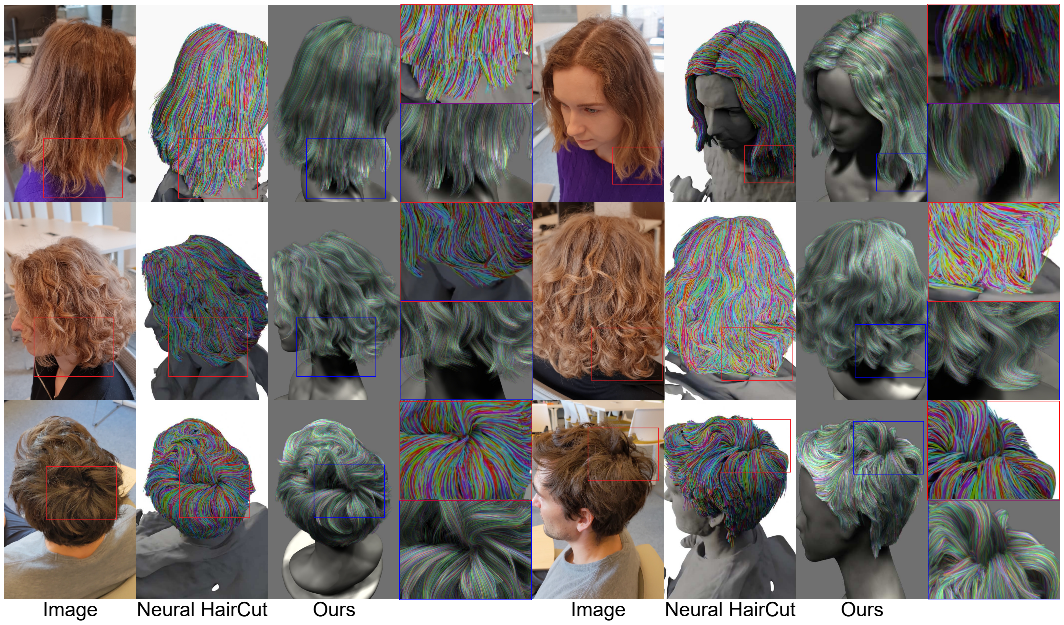


Figure 13. Extended qualitative comparison with Neural Haircut [30].

Method	Thresholds: mm / degrees								
	2/20	3/30	4/40	2/20	3/30	4/40	2/20	3/30	4/40
	Precision			**Recall**			**F-score**		
w/o PMVO	30.3	54.5	67.3	6.7	14.5	21.6	11.0	22.9	32.7
w/o DeepMVSHair*	**65.2**	**87.4**	**95.6**	6.3	15.2	19.8	11.5	25.9	32.8
Full	60.8	83.3	92.1	**10.4**	**19.3**	**25.9**	**17.8**	**31.3**	**40.4**

Table 3. Quantitative evaluation of individual components of our method on the synthetic dataset [41].

ducted two sets of ablation studies: 1) without the undirectional strand map (w/o Map) and 2) without the proposed new loss function $\mathcal{L}_{ori}$. The comparison results are shown in Fig. 15. With no undirectional strand map, the reconstructed results are more susceptible to the loss of geometric details due to ambiguity. While the loss function $\mathcal{L}_{ori}$ helps us train DeepMVSHair* more robustly.

Comparison with different patch sizes. We also compared the effects of different sample sizes (patch size) in PMVO on the reconstruction of the exterior hair geometry. As shown in Fig. 16 and Tab. 4, setting the patch size to 5 better balances performance and efficiency.

7.3. More results

Lastly, we provide additional reconstruction examples from a monocular video showcasing various hairstyles such as short, long, straight, curly, and wavy hair. As shown in Figs. 17 to 19, MonoHair can handle diverse, complex hairstyles and achieve high-fidelity results.

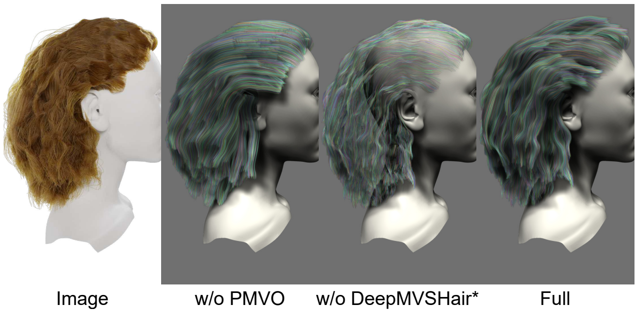


Figure 14. Qualitative evaluation of each key component of our method on synthetic data.

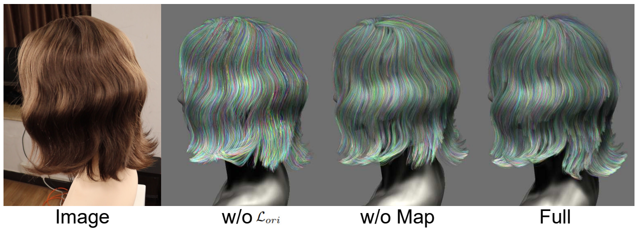


Figure 15. Qualitative evaluation of the improvements to DeepMVSHair.

Patch size	1	3	5	7
Time	∼30min	∼ 90min	3h-4h	6h-8h

Table 4. Comparison of time consumption under different patch sizes. As the patch size increases, the time consumption increases significantly.

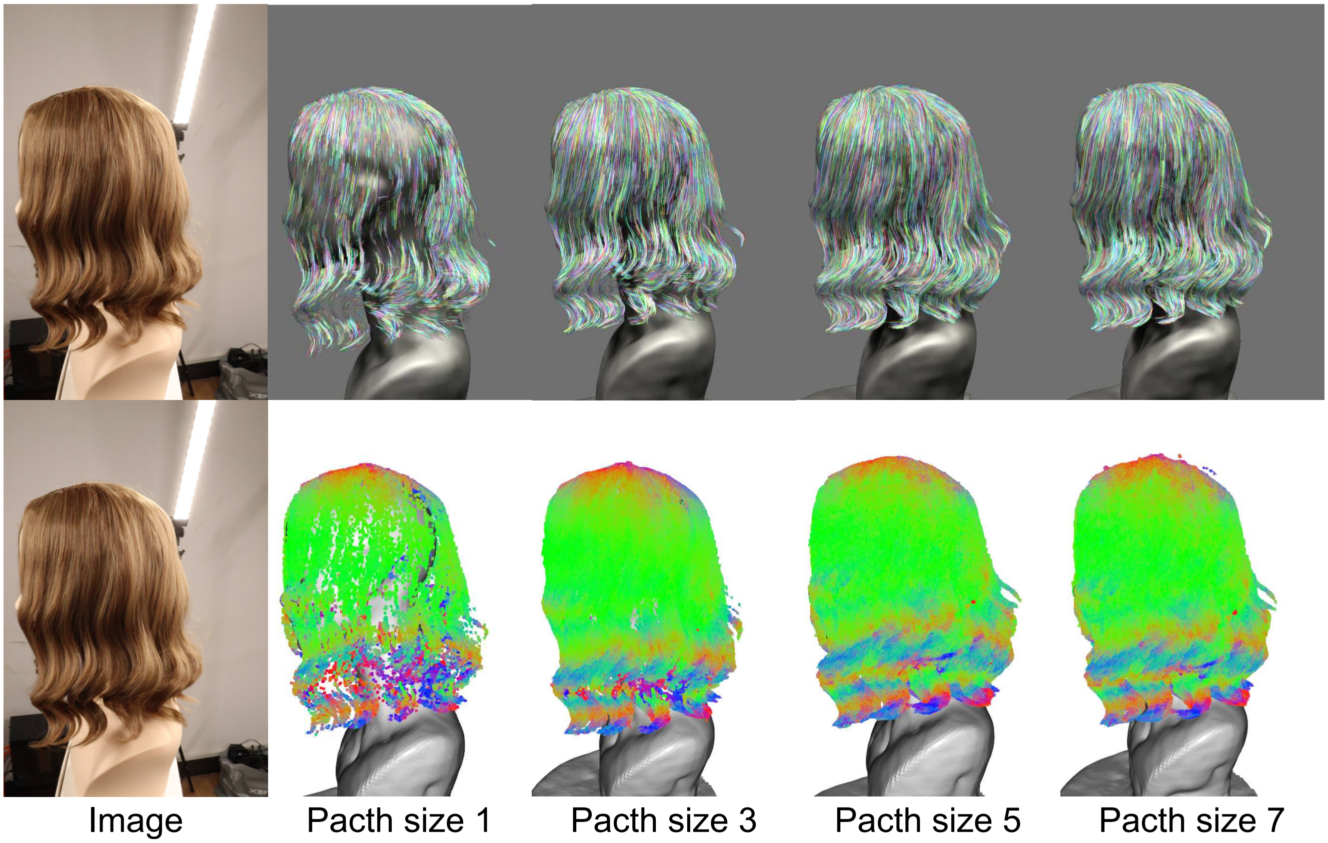


Figure 16. Qualitative comparison of PMVO with different patch sizes. When the patch size is greater than 5, there is no significant improvement in the reconstructed geometry. The above results are all the exterior geometry of the hair.

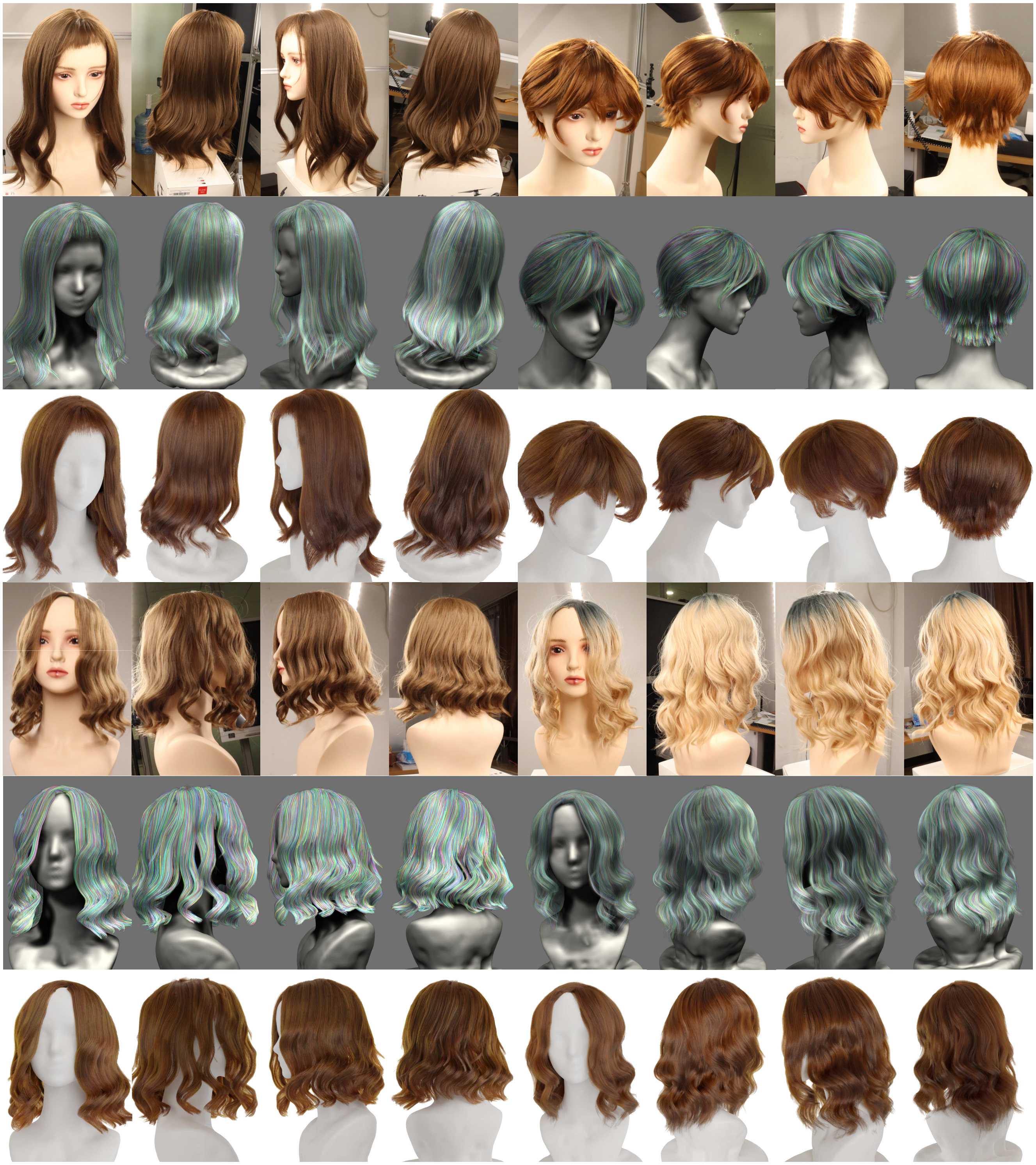


Figure 17. Additional reconstruction results of our method based on monocular videos. We use two rendering styles to showcase our results: colorful strands for displaying hair geometry and Blender [5] for realistic rendering.

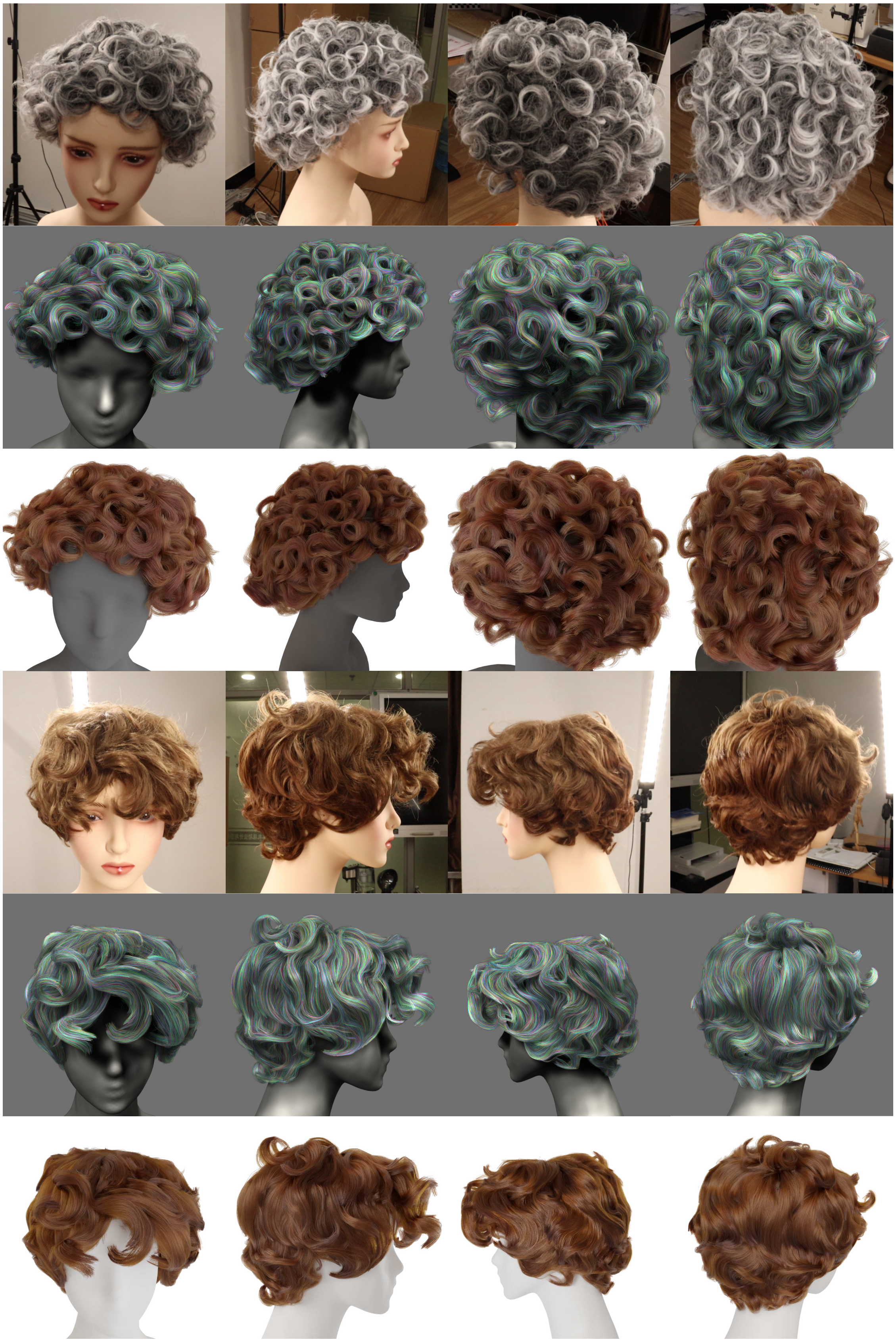


Figure 18. Additional reconstruction results of our method based on monocular videos.

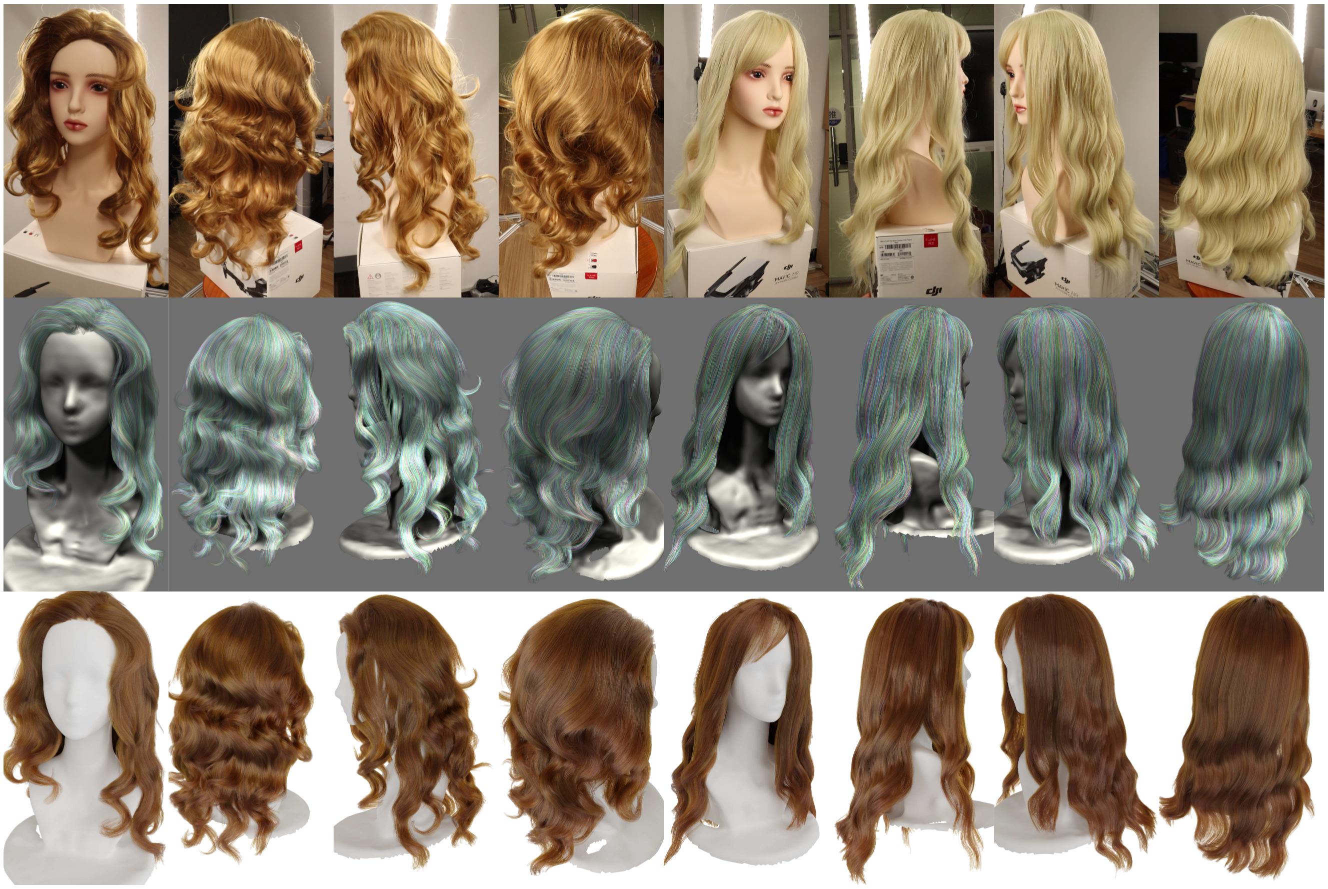


Figure 19. Additional reconstruction results of our method based on monocular videos.