DeepMVSHair: Deep Hair Modeling from Sparse Views

Zhiyi Kuang
musinghead@zju.edu.cn
Zhejiang University
China

Yiyang Chen
chen_yy@zju.edu.cn
Zhejiang University
China

Hongbo Fu
hongbofu@cityu.edu.hk
City University of Hong Kong
China

Kun Zhou
kunzhou@acm.org
Zhejiang University
China

Youyi Zheng*
youyizheng@zju.edu.cn
Zhejiang University
China

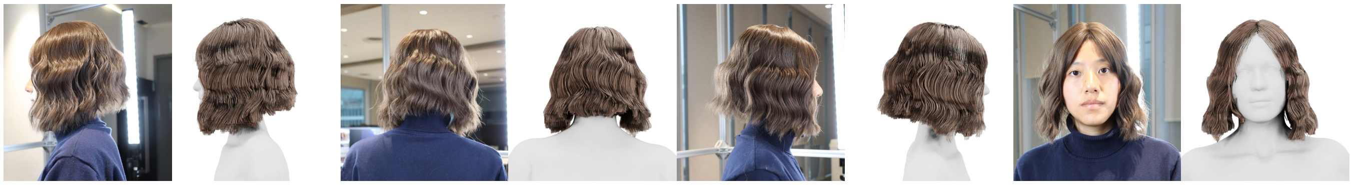

Figure 1: Given sparsely captured portrait images, our DeepMVSHair automatically produces a complete hair strand model accurately matching all views. The key component of our pipeline is HairMVSNet, a neural architecture efficiently gathering hair structure features from multiple input views. The displayed 3D bust model is fitted using [Cao et al. 2014].

ABSTRACT

We present DeepMVSHair, the first deep learning-based method for multi-view hair strand reconstruction. The key component of our pipeline is HairMVSNet, a differentiable neural architecture which represents a spatial hair structure as a continuous 3D hair growing direction field implicitly. Specifically, given a 3D query point, we decide its occupancy value and direction from observed 2D structure features. With the query point's pixel-aligned features from each input view, we utilize a view-aware transformer encoder to aggregate anisotropic structure features to an integrated representation, which is decoded to yield 3D occupancy and direction at the query point. HairMVSNet effectively gathers multi-view hair structure features and preserves high-frequency details based on this implicit representation. Guided by HairMVSNet, our hair-growing algorithm produces results faithful to input multi-view images. We propose a novel image-guided multi-view strand deformation algorithm to enrich modeling details further. Extensive experiments show that the results by our sparse-view method are comparable to those by state-of-the-art dense multi-view methods and significantly better than those by single-view and sparse-view methods. In addition, our method is an order of magnitude faster than previous multi-view hair modeling methods.

CCS CONCEPTS

• **Computing methodologies** → Multi-view stereo; Neural networks; Implicit functions.

KEYWORDS

hair modeling, neural networks, implicit functions

ACM Reference Format:
Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. 2022. DeepMVSHair: Deep Hair Modeling from Sparse Views. In *Special Interest Group on Computer Graphics and Interactive Techniques in Asia Conference Proceedings (SIGGRAPH Asia'22 Conference Proceedings), December 6-9, 2022, Daegu, South Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3550469.3555385

1 INTRODUCTION

Hair modeling is a significant component of human digitalization, which has been actively studied with the rapid development of VR and AR applications. While face modeling [Cao et al. 2015] has gained great success, due to the complex nature of hair strands, high fidelity and efficient hair modeling techniques have not been well developed, making hair modeling a bottleneck to modeling realistic virtual humans.

Existing image-based hair modeling methods have achieved impressive results but compromise on at least one of various aspects including data capturing complexity, processing time, manual operation, or modeling quality. Dense multi-view hair modeling techniques [Hu et al. 2014; Luo et al. 2013a; Nam et al. 2019; Paris et al. 2008] currently produce state-of-the-art modeling quality with a dense camera array, controlled lighting, and long processing time. Such requirements keep these methods far from average users and efficient usage. Techniques using less constrained inputs, such as sparse views [Zhang et al. 2017], selfie videos [Liang et al. 2018], and RGBD streams [Zhang et al. 2018], are more efficient to use.

[*] corresponding author

SIGGRAPH Asia'22 Conference Proceedings, December 6-9, 2022, Daegu, South Korea
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3550469.3555385

However, their results fall short of exact details because they utilize shape priors (such as a hair database and user strokes) to integrate hair structures from different input views. Single-view hair modeling techniques [Chai et al. 2016, 2013; Hu et al. 2015; Saito et al. 2018; Yang et al. 2019; Zhang and Zheng 2018; Zhou et al. 2018] use only a front face image to produce plausible results, which, however, only resemble visible frontal parts of target hair and are not sufficient for complete human digitalization.

To achieve high-fidelity and efficient usage for hair modeling, we introduce DeepMVSHair, a deep learning-based method for multi-view hair modeling. The key component of our method is HairMVSNet which innovatively represents a spatial hair structure as a purely implicit direction field. This implicit representation is intrinsically continuous and can preserve more high-frequency details than explicit voxels used in previous learning-based hair modeling methods. Instead of using sparse views to optimize a shape prior as in [Zhang et al. 2017], HairMVSNet infers 3D hair occupancy and growing direction at a query point from its observed 2D pixel-aligned features at each input view. This formulation encourages HairMVSNet to learn to aggregate local features rather than to predict a global shape, making HairMVSNet generalizable to unseen hairstyles and possess rich details. These 2D pixel-aligned features from multiple views are aggregated by a view-aware transformer encoder and decoded by an MLP to predict hair occupancy and growing direction at the query point. Guided by HairMVSNet, we grow hair strands resembling all input views. To increase modeling fidelity further, we design a multi-view strand deformation algorithm, revisiting input views to fine-tune hair strands with 2D guidance to match all views consistently.

As a result, we make a minimal overall compromise on the aspects mentioned above: 1) Data capturing complexity. We only require sparse views that roughly cover most parts of hair geometry. Moreover, we use 2D direction maps, which represent hair structures rather than color images as input, allowing our method to work with casual lighting without uniformly controlled lighting required in traditional MVS methods. 2) Processing time. We typically take around 1 minute (depending on the size of hair volume) to generate a complete hair strand model, while classical stereo matching-based multi-view methods require tens of minutes to hours. 3) Manual operation. Our pipeline is fully automatic. 4) Modeling quality. Demonstrated by experiments on challenging complex hairstyles, using only sparse input views, our method achieves high quality comparable to state-of-the-art dense multi-view methods and significantly outperforms existing single-view and sparse-view methods. Also, the lightweight nature of our method enables our system to be easily deployed for fast avatar creation in many life sites such as shopping malls, VR/MR experience stores, fashion salons, and exhibition halls.

In summary, our contributions are as follows:

- We introduce the first deep learning-based method for multi-view strand hair modeling, using only sparse views to reconstruct complete hair geometry.
- We introduce HairMVSNet, which employs an implicit representation of the hair-growing field to efficiently aggregate hair structure features from multiple views.
- We introduce a novel image-guided multi-view strand deformation algorithm to refine modeling results.
- Our method outperforms existing single-view or sparse-view hair modeling methods, and achieves comparable modeling quality to state-of-the-art dense-view methods with an order of magnitude faster inference performance.

Method	Views	Range	Auto.	Time
[Yang et al. 2019]	Single	Front	✓	~15s
[Wu et al. 2022]	Single	Front	✓	~10s
[Zhang et al. 2017]	Sparse	Complete	×	~25m
[Hu et al. 2014]	Dense	Complete	✓	1-2h
Ours	Sparse	Complete	✓	~1m

Table 1: Taxonomy of methods. Views: number of input views. Range: faithfully modelled areas of target hair. Auto: whether fully automatic. Time: processing time per case.

2 RELATED WORK

Multi-view Hair Modeling. Multi-view stereo has been long studied to reconstruct 3D geometry from a set of captured images. Due to the thin and complex nature of hair strands, MVS requires additional adaptation to be applied to hair modeling. Luo et al. [2012; 2013b] employ hair orientations as constraints to deform a hair mesh to produce more high-frequency details. Paris et al. [2008] use a 3D orientation triangulation technique to recover a 3D growing volume to generate strands. The works of [Hu et al. 2014; Luo et al. 2013a] utilize shape primitives (ribbons, wisps, strands) to fit an original point cloud produced by MVS to generate complete connect-to-scalp hair strands. Nam et al. [2019] introduce a line-based PatchMatch MVS, designed to reconstruct thin strand segments. Sun et al. [2021] utilize a per-pixel lightcode to boost the stereo matching process and estimate hair reflectance properties for realistic rendering. Dense MVS-based methods currently produce state-of-the-art hair modeling quality. However, they require a dense camera array, uniform lighting control, and long processing time (tens of minutes to hours), thus making their application scenarios far from average users.

To simplify the data capturing process, several works try modeling hair from less constrained inputs, such as sparse views [Zhang et al. 2017], selfie videos [Liang et al. 2018], and RGBD streams [Zhang et al. 2018]. These methods produce plausible results with the help of either user interactions or a hair database, but fall short of faithful details to original input views.

Single-view Hair Modeling. Another research direction is to take a front face image as input and generate hair strands matching frontal visible hair. Chai et al. [2015; 2016; 2013; 2012] propose a series of single-view hair modeling methods, stepping forward to automatic hair modeling. Hu et al. [2015] combine exemplar hair models from a database to match a reference image with a few user interactions. Deep learning based methods [Saito et al. 2018; Yang et al. 2019; Zhang and Zheng 2018; Zhou et al. 2018] generate plausible results automatically with fast inference. However, their explicit hair representation (hair strand vertices, orientation voxels)

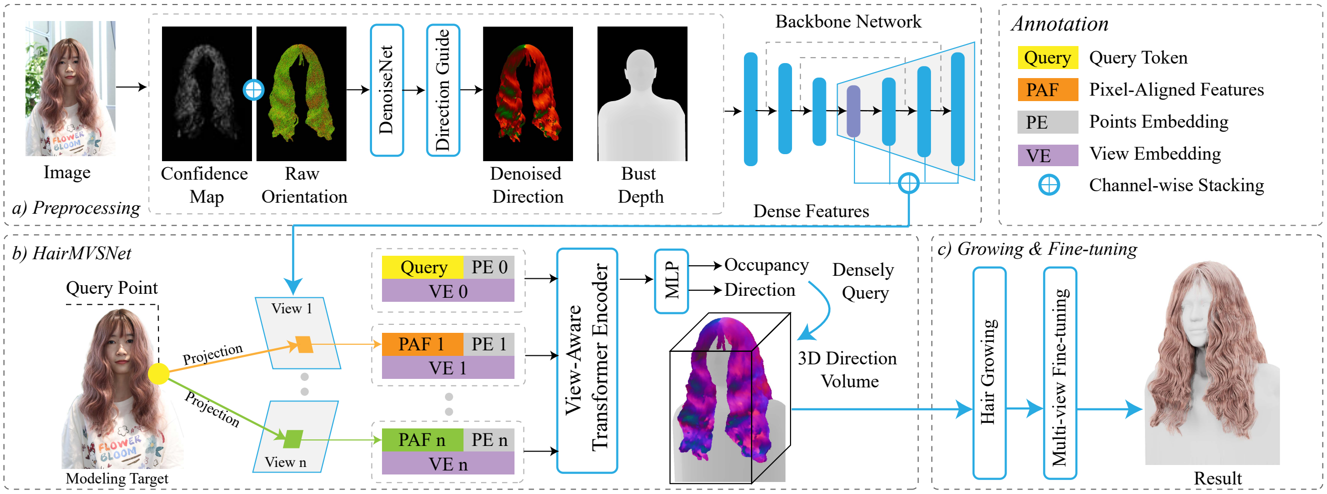

Figure 2: Overall pipeline. (a) Each captured image is processed to yield a denoised 2D direction map and a bust depth map. These two maps are then fed to a backbone network to produce a deep feature map. (b) HairMVSNet effectively integrates multi-view hair structure features to infer each query point's 3D occupancy and direction, forming a hair growing direction volume. (c) Hair strands are grown and fine-tuned in the hair volume to produce final results.

limits both resolution and high frequency details. Single-view hair modeling methods work well for frontal avatar generation but are not sufficient for complete human digitalization since their results fail to match target hair at views distant from the input image.

Neural Implicit Representation. Implicit representations use continuous functions to encode 3D spatial features, such as occupancy value [Mescheder et al. 2019] and signed distance [Park et al. 2019] to object surfaces. Due to their differentiable and compact nature, they have received extensive attention in 3D modeling and shown promising results. For example, NeRF [Mildenhall et al. 2020] learns a volumetric representation of scenes, where an RGB color field and a density field are encoded as a coordinate-based function. DVR [Niemeyer et al. 2020] and IDR [Yariv et al. 2020] use differentiable rendering to optimize implicit surfaces. Unisurf [Oechsle et al. 2021] progressively converges sampling regions close to object surfaces, thus transforming rough volumes to actual surfaces. NeuS [Wang et al. 2021a] transforms an SDF to a volume density field by proposing an unbiased weight function to integrate colors of sampling points. Wang et al. [2021b] represent hair appearance as a combination of shape primitives and an implicit texture field to capture dynamic hair performance. Besides per-scene representation training, pixel-aligned features [Huang et al. 2018] [Saito et al. 2019] are introduced for implicit representation inference. A concurrent work [Wu et al. 2022] has applied implicit representation and pixel-aligned features to the single-view hair modeling task. Different from their hybrid representation including an implicit direction field and explicit voxels as intermediate, we propose a purely implicit representation for hair structures and use a view-aware transformer encoder to successfully aggregate multi-view information, which is necessary to produce results matching all input views rather than only the front view.

3 METHODOLOGY

Our method takes sparse-view images with calibrated camera intrinsics and extrinsics as input to automatically produce high-fidelity hair strands matching all the input views. As shown in Fig. 2, our pipeline is threefold. We first process each captured image to a denoised 2D direction map and a bust depth map, which are stacked channel-wise and fed to a backbone network to produce a deep feature map (Sec. 3.1). Then we densely sample points in the space around the subject in the images and query their occupancy and direction values through HairMVSNet (Sec. 3.2) to form a hair growing volume. HairMVSNet is trained on synthetic data since there is no effective way to capture ground-truth 3D hair growing directions inside hair volumes. Finally we grow and fine-tune strands in this volume to generate high-fidelity results (Sec. 3.3).

3.1 Preprocessing

Denoised Direction Maps. We first segment the hair regions using the method proposed in [Chai et al. 2016]. Then, to fill the gap between synthetic and real hair images, we estimate 2D orientation maps to represent hair structures and 2D confidence maps to measure orientation accuracy following [Paris et al. 2004]. Orientation at low-confidence regions is noisy and less accurate, which has long been a bottleneck of hair modeling from average quality images. To overcome this limitation, we train DenoiseNet, which utilizes a U-Net structure to infer accurate orientation from raw orientation and confidence maps, as shown in Fig. 2a. DenoiseNet is trained with raw orientation and confidence maps extracted from rendered synthetic hair and paired ground-truth orientation maps (we refer to our supplementary document for the evaluation of DenoiseNet). We then use the method proposed in [Chai et al. 2016] to obtain direction label maps and combine them with the denoised orientation maps to remove directional ambiguity and produce the denoised direction maps.

Bust Depth Maps. We use a face tracking approach [Cao et al. 2014] to fit a general bust model to the subject. Then we render depth maps of the bust at all views. These depth maps serve as clues of the internal body volume to help HairMVSNet learn body-hair occlusions to focus on modeling hair regions.

3.2 HairMVSNet

Architecture. The goal of HairMVSNet is to learn a general feature aggregation mapping from a query point's observed 2D hair structure features at multiple views to its 3D occupancy and growing direction. This general mapping should preserve high-frequency details and be applicable to unseen real hairstyles. To fulfill these two requirements, we model HairMVSNet as a neural implicit field, which is free of an explicit resolution limit to achieve high accuracy and learns generic local hair structure patterns (instead of global hair shapes) to be well transferred to unseen hairstyles. HairMVSNet is formulated as:

$$H(X, \{Dir, Dep\}_1, ..., \{Dir, Dep\}_n) = (\sigma, d), \quad (1)$$

where HairMVSNet H takes the denoised 2D direction maps Dir and the bust depth maps Dep and a 3D query point X as input to predict its occupancy σ and 3D direction d.

To start with, we stack Dir and Dep of input view i channel-wisely and feed them to a backbone network F to produce a deep feature map f_i

$$f_i = F(\{Dir, Dep\}_i). \quad (2)$$

There is no specific limitation on the choice of the backbone network F and any dense prediction network should suffice. We found that a lightweight U-Net strikes a good balance between inference accuracy and efficiency.

We then fetch pixel-aligned features $f_i(x_i)$ of query point X from each captured view. Unlike isotropic features such as texture colors, hair growing directions are anisotropic and show different 2D observations at different views. Therefore, besides pixel-aligned features, view-aware features are necessary to inform HairMVSNet of query points' observation views. A complete feature token ϕ_i from an input view is formulated as:

$$\phi_i = g(f_i(x_i), p_i(X)) + e_i, \quad (3)$$

where $g(\cdot)$ is an MLP to fuse features. The pixel-aligned features are augmented by the view-aware features, including the query point X's position $p_i(X)$ at the observation camera's coordinate and a learnable view embedding e_i. These view-aware features help to learn correlations between views and significantly improve modeling accuracy both qualitatively (Fig. 4) and quantitatively (Tab. 2).

To integrate feature tokens from multiple views, the integration method is required to work with an unordered and arbitrary number of input feature tokens. A transformer [Vaswani et al. 2017] encoder is suitable for this task:

$$E\left(q(X), \phi_1, ..., \phi_n\right) = \Phi_X, \quad (4)$$

where a query token q fused with the position embedding of X is employed to query feature tokens $\{\phi_i\}$ from multiple views to generate an integrated feature token Φ_X. The transformer encoder E is aware of each feature token's view identity with view-aware features and uses stacked attention blocks to effectively correlate view feature tokens.

Finally, we decode Φ_X using an MLP to predict occupancy σ and direction d of the query point X:

$$MLP(\Phi_X) = (\sigma, d). \quad (5)$$

Loss Functions. The predicted occupancy and 3D direction values are supervised by ground-truth synthetic data. For each training pass, we feed HairMVSNet with a batch of N sampling points to compute gradients. Occupancy is formulated as a binary classification problem and trained with the cross entropy loss:

$$L_{occ} = \frac{1}{N}\sum_{i}^{N}\left[\sigma^* \log\sigma + (1-\sigma^*)\log(1-\sigma)\right], \quad (6)$$

where σ^* is the ground-truth occupancy value.

3D direction is trained with the average L-1 loss of vector components on each axis. Specifically, we have

$$L_{ori} = \frac{1}{N}\sum_{i}^{N}\frac{\|d^* - d\|_1}{3}. \quad (7)$$

Training Data. We use 343 synthetic hairstyles from the USC-HairSalon dataset introduced in [Hu et al. 2015]. Diverse hairstyles (including long, short, straight, and curly hairstyles) are used to train a generalizable model. Each synthetic training case is composed of:

1) Sparse view images. To make camera views roughly cover the whole hair model without loss of generality, we initially set virtual cameras uniformly distributed on a circle around a hair model and then add random variations to both camera poses and intrinsics for each training case.

2) Sampling points. For occupancy prediction, since hair strands physically occupy very few volumes, we voxelize the 3D space around a hair model and regard a voxel as positive if it is crossed by any strands, otherwise as negative. Then we densely sample points at both positive voxels and their neighboring negative voxels, and sparsely sample points in the remaining empty space. This setting encourages our learning to focus on hair regions and balances positive and negative samples. For direction prediction, we use original hair strand vertices as sampling points to maintain high resolution.

We augment our synthetic dataset with random scaling, rotation and translation on hair models to obtain 2,195 training cases and 549 test cases.

3.3 Strands Generation and Refinement

Growing Volume Construction. To generate hair strands efficiently, we build a hair-growing volume by voxelizing the space around the subject in the input images and querying occupancy and direction values at voxel corners through HairMVSNet. The resolution and range of voxels are free of an explicit limit based on our implicit representation. We found that a setup of 5mm voxel side length and 0.5m × 0.6m × 0.8m (L × D × H) covering range strikes a good balance between querying efficiency and volume resolution for all of our captured data.

Hair Growing. We uniformly sample starting points and grow strands bidirectionally in the growing volume. In each growing step, the growing direction of the current strand end is a trilinear interpolation result of directions at the corners of the voxel that contains the strand end.

Multi-view Strand Deformation. Although the hair strands initially grown in the hair-growing volume already resemble all input views, they might be over-smooth. This is because in practice the resolution of the hair-growing volume cannot be infinitely high, which impairs the efficiency of our pipeline. Therefore, with the key insight that a strand should match its 2D observation from views where it is visible, we revisit the input images and devise an image-guided strand deformation to fine-tune the modeling result.

As shown in Fig. 3, for each input image, we first trace strands along its 2D direction map inside the hair regions to form a set of 2D strands $G = \{g\}$ (Fig. 3b). These 2D strands possess most details in the input image (Fig. 3a) and serve as guidance for deformation. We then project each 3D strand $\hat{s}$ to views where it is visible to yield a set of projected 2D strands $S = \{s\}$ (Fig. 3c). Each projected 2D strand s is deformed w.r.t. its matching guide strands (Fig. 3d blue) from G in a segment-wise manner to generate a set of deformed 2D strands $T = \{t\}$, and this deformation is resistant to crossing strands (Fig. 3d red) that do not match s. The strands in T are integrated to obtain the final deformed 3D strand $\hat{t}$ by unprojection (Fig. 3e). Our deformation algorithm ensures consistency across different views, as the fine-tuned result of a real hairstyle shown in Fig. 4f (fine-tuned on Fig. 4e). We refer to our supplementary document for more details of our deformation algorithm and a comparison with a single-view deformation method proposed in [Hu et al. 2015].

4 EXPERIMENTS

4.1 Capture System

We use 4 Canon EOS 850d cameras mounted on a box frame structure to capture image data. When an actor/actress sits at the center of the box, these 4 cameras are approximately set at his/her front, back, left, and right, respectively to roughly cover most of the hair geometry. Camera intrinsics and extrinsics are calibrated with a checkerboard pattern. The camera captures are synchronized with remote cable releases. 4 LED lights are placed around the box frame to provide a bright environment for a better capture of hair texture. Note that we do not require a uniform lighting condition, and thus casual light sources providing properly bright illumination should suffice.

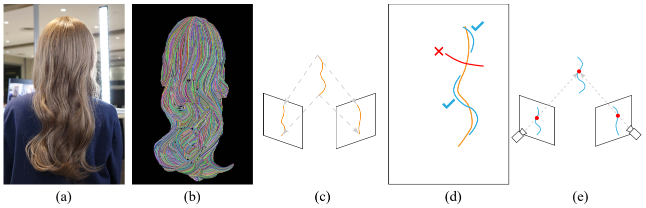

Figure 3: Image-guided deformation. (a) Input Image. (b) 2D guide strands. (c) Project a 3D strand to the camera views where it is visible. (d) Deform a projected strand (orange) w.r.t its matching guide strands (blue) on a 2D plane. (e) Unproject the deformed 2D strands to 3D space.

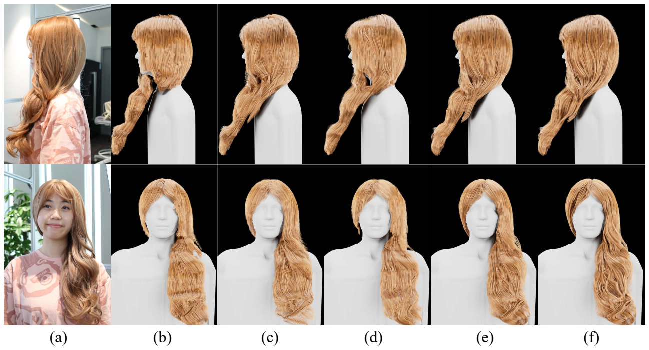

Figure 4: Qualitative comparisons of feature aggregation methods including average pooling (b), MLP mapping (c), ours without view-aware features (d), and ours (e). (f) is our fine-tuned (Sec. 3.3) full result and (b)-(e)are not fine-tuned for a fair comparison.

Method	Precision (%) ↑	Recall (%) ↑	Dir. L1 Loss ↓
AVG	82.27	84.95	0.1679
MLP	84.07	83.76	0.1733
Ours w/o VA	83.27	84.18	0.1725
Ours	**84.08**	**85.96**	**0.1295**

Table 2: Quantitative comparisons of feature aggregation methods. AVG: average pooling; VA: view-aware features. Our method infers directions with a significantly lower loss.

4.2 Evaluation

Evaluation of View-aware Transformer Encoder. To validate the effectiveness of our view-aware transformer encoder, we conduct an ablation study by replacing it with other feature aggregation methods, including average pooling and MLP mapping with the same view-aware feature tokens as input. We also evaluate our transformer encoder's performance without the view-aware features. As shown in Tab. 2, although the other aggregation methods achieve comparable precision and recall of occupancy inference on the synthetic dataset, they fall short of inferring accurate directions due to the anisotropic nature of directions. When transferring to real data shown in Fig. 4, the other methods produce fewer details compared to ours and cause discontinuity in the hair volume. This is because the hairstyle shown in Fig. 4a mainly distributes on the left side of the actress and the other methods are disturbed by empty 2D observations from other input views, while our method properly integrates unevenly distributed hair structures.

Evaluation of Various Numbers of Views. Although we capture real hairstyles using only 4 cameras in our experiments (we assume 4 views to be minimal to cover a complete hair), our pipeline is capable to accept more than 4 input views to achieve better modeling accuracy. We evaluate our pipeline's performance with the increased number of input views on the synthetic dataset. In this evaluation, a shallow convolutional network is used as the backbone network to avoid lack of graphics memory when more images

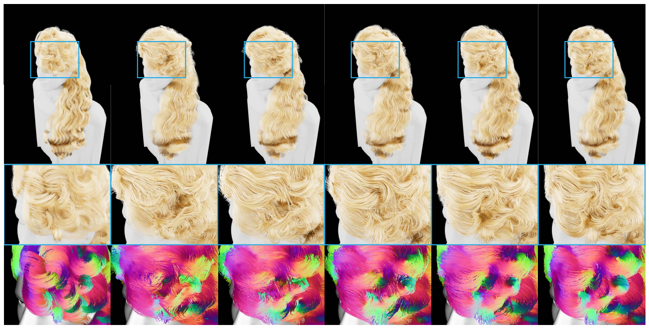

Figure 5: Comparisons of results using different numbers of input views. From left to right: ground truth, respective results using 4, 8, 12, 16, and 24 views. The second row displays zoom-in views and the third row displays hair-growing directions (encoded in color).

Num. of Views	Precision (%) ↑	Recall (%) ↑	Dir. L1 Loss ↓
4	81.68	86.61	0.1326
8	83.91	85.62	0.1318
12	84.53	88.06	0.1265
16	86.32	88.48	0.1186
24	**87.02**	**89.21**	**0.1158**

Table 3: Quantitative comparisons of different numbers of input views. More input views contribute to higher accuracy.

are added (this is why the results using 8 views shown in Tab. 3 are less accurate than those using 4 views shown in Tab. 2). As shown in Tab. 3 and Fig. 5, both occupancy and direction become more accurate with more input views, showing that our pipeline effectively learns the correlation between views.

4.3 Comparisons

We first compare our method with two single-view deep learning-based hair modeling methods: [Yang et al. 2019] and [Wu et al. 2022]. The former is based on a volumetric representation while the latter uses implicit fields with voxels as an intermediate representation. It can be seen from Fig. 6 that our method, which uses a purely implicit function as our neural architecture, produces a result with richer details faithful to the input image. Quantitatively evaluated on the 343 synthetic hairstyles used in our experiments (also included in the training dataset of [Wu et al. 2022]), we achieve 84.08% precision of occupancy and 0.1295 L1 loss of direction (as shown in Tab. 2), better than 76.36% precision and 0.1750 L1 Loss achieved by [Wu et al. 2022]. This demonstrates that our method properly integrates pixel-aligned features from multiple input views to achieve better accuracy.

We then compare our method with a state-of-the-art sparse view hair modeling method [Zhang et al. 2017]. Since they deform an exemplar hair model to match observed hair shapes, their results are coarser (especially at thin tail parts) than ours, as shown in Fig. 7. In addition, their method involves several manual operations and takes around 25 minutes to generate a hair strand model, while our method takes only 1 minute without any manual labor.

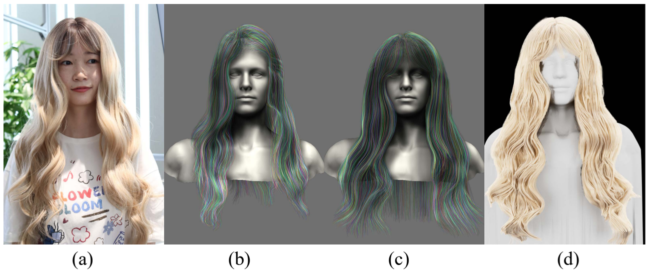

Figure 6: Qualitative comparison between our method (d) and two single-view hair modeling methods: [Yang et al. 2019] (b) and [Wu et al. 2022] (c).

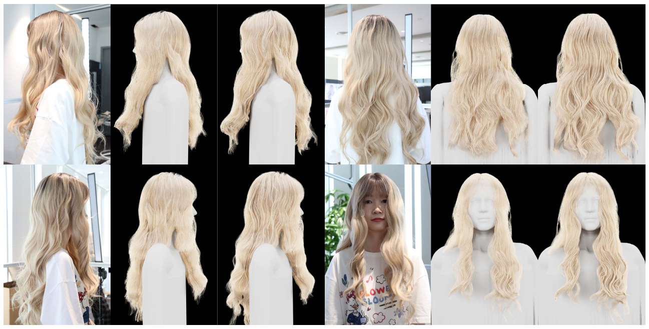

Figure 7: A comparison with a state-of-the-art sparse-view hair modeling method. From left to right: input images, results using [Zhang et al. 2017], and ours.

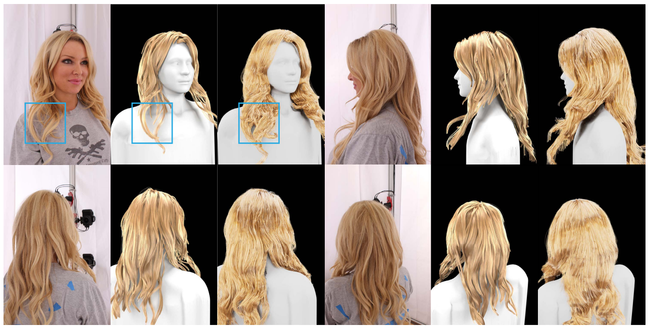

Figure 8: A comparison with a state-of-the-art dense-view hair modeling method. From left to right: input images, results using [Hu et al. 2014], and ours.

We also compare our method with a state-of-the-art dense view hair modeling method [Hu et al. 2014]. They produce an accurate strand model with 66 input images based on traditional MVS and an exemplar strands fitting algorithm with 1-2 hours processing time. We use only 8 input views (selected to cover most of the hair geometry) to produce results also matching all input views, though some empty spaces between thin strands are not accurately

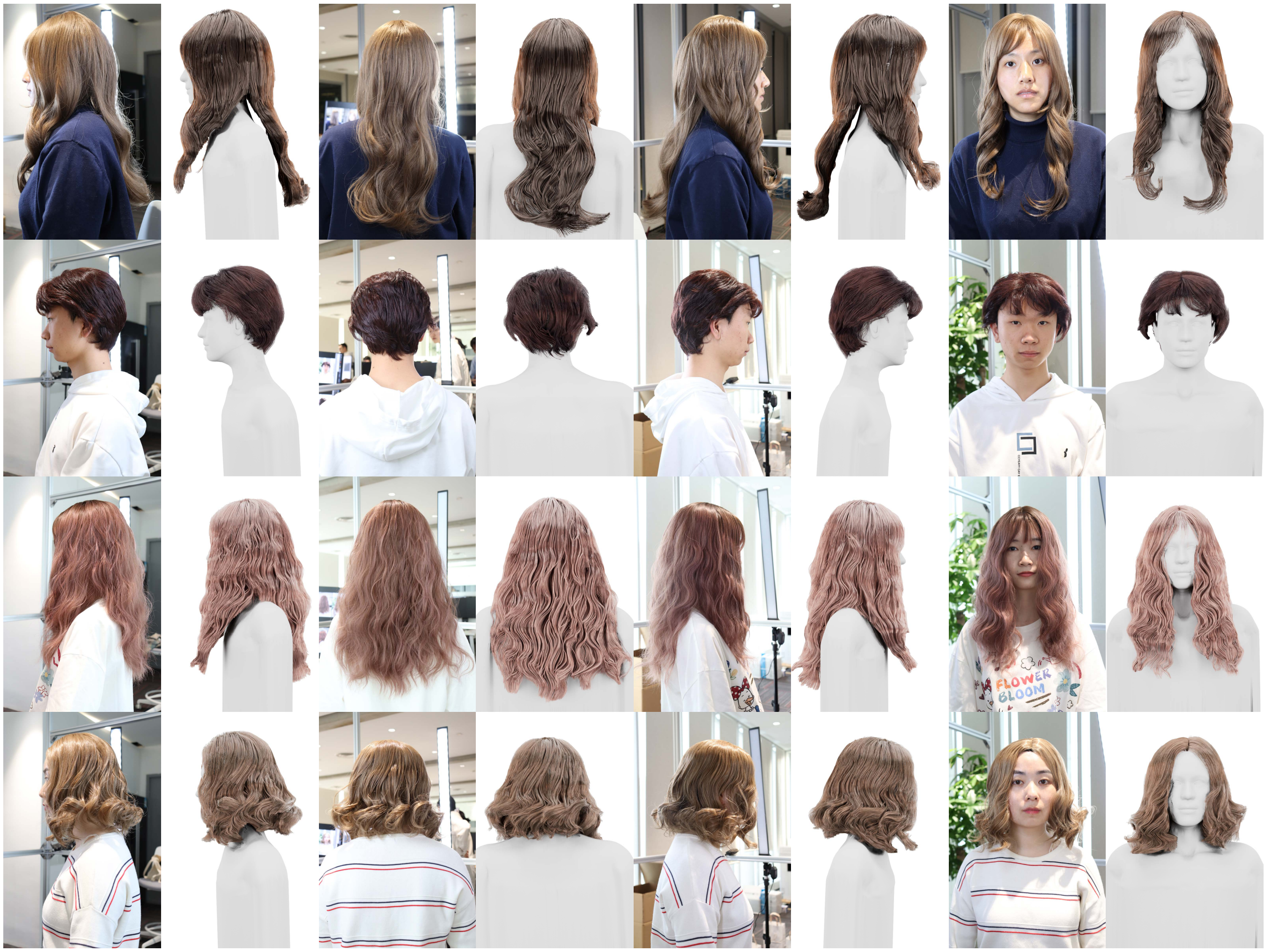

Figure 9: Our results of various hairstyles, rendered at input views and displayed side by side with the input images.

recovered (Fig. 8 blue square) because the deficiency in the segmentation method we used (a continuous and inaccurate mask is predicted). Note that our pipeline only takes around 1 minute to generate full results, making a step forward to efficient high quality hair modeling.

5 DISCUSSIONS AND LIMITATIONS

Although our method can faithfully model various hairstyles, there are still a few challenges to overcome, which may inspire future work. First, our method cannot model complex internal structures (such as braids) because they are not visible from input images. Also, since our method takes the hair-growing orientation as input, it may fail to model extremely short or curly (such as an Afro) hairstyles if their orientations cannot be well extracted from images. To improve our system, strand shape priors might be utilized to supplement plausible structures when observed orientations are insufficient. Second, we only capture hair strand geometry, while texture and material properties are also essential for realistic hair. Incorporating these properties into synthetic hair datasets should be helpful to infer properties of real hairstyles under less-constrained capture setting. Third, with proper adaptation, a more convenient capture procedure such as recording a video using smartphones (camera poses can be measured by built-in IMUs) might promote our technique to individual users.

6 CONCLUSION

We have proposed a deep learning-based method for multi-view hair modeling, enabling high-quality hair modeling to be performed efficiently. We propose DenoiseNet to infer accurate hair orientation maps from raw orientation maps, since the lack of accurate hair orientation maps has long been a bottleneck in applying hair modeling methods to average quality images. We also propose an image-guided strand deformation algorithm to increase modeling fidelity further. Our method works robustly on a wide range of hairstyles with only sparse view inputs.

ACKNOWLEDGMENTS

We thank the reviewers for their constructive comments. This work was supported in part by the NSF China (No. 62172363). Hongbo Fu was supported by a grant from the City University of Hong Kong (Project No. 7005729).

REFERENCES

Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. 2015. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–9.

Chen Cao, Qiming Hou, and Kun Zhou. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)* 33, 4 (2014), 1–10.

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–10.

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

Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. 2013. Dynamic hair manipulation in images and videos. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–8.

Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. 2012. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8.

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

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

Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. 2018. Deep volumetric video from very sparse multi-view performance capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 336–354.

Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. 2018. Video to fully automatic 3d hair model. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.

Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. 2012. Multi-view hair capture using orientation fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1490–1497.

Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013a. Structure-aware hair capture. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

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

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*. Springer, 405–421.

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

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3504–3515.

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

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

Sylvain Paris, Will Chang, Oleg I Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.* 27, 3 (2008), 30.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 165–174.

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

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2304–2314.

Tiancheng Sun, Giljoo Nam, Carlos Aliaga, Christophe Hery, and Ravi Ramamoorthi. 2021. Human Hair Inverse Rendering using Multi-View Photometric data. (2021).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021a. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021).

Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Michael Zollhoefer, Jessica Hodgins, and Christoph Lassner. 2021b. HVH: Learning a Hybrid Neural Volumetric Representation for Dynamic Hair Performance Capture. *arXiv preprint arXiv:2112.06904* (2021).

Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. 2022. NeuralHDHair: Automatic High-fidelity Hair Modeling from a Single Image Using Implicit Neural Representations. https://doi.org/10.48550/ARXIV.2205.04175

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

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems* 33 (2020), 2492–2502.

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

Meng Zhang, Pan Wu, Hongzhi Wu, Yanlin Weng, Youyi Zheng, and Kun Zhou. 2018. Modeling hair from an rgb-d camera. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Meng Zhang and Youyi Zheng. 2018. Hair-gans: Recovering 3d hair structure from a single image. *arXiv preprint arXiv:1811.06229* (2018).

Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. 2018. Hairnet: Single-view hair reconstruction using convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 235–251.